DISCOVER THE WORLD OF RASPBERRY PI! **NEW MAG!**

PiUser

Issue 01 // Winter 2016

YOUR NEW SOURCE FOR RASPBERRY PI
JOIN THE PI
REVOLUTION

EVERYTHING YOU NEED TO POWER UP YOUR RASPBERRY PI!

DISPLAY
PWR IN
HDMI
CAMERA
GPIO
BROADCOM
Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
RUN
USB

11 AMAZING Pi PROJECTS

Build a Pi drone

Step-by-step guide to making your own drone!

Make a retro arcade classic

Games are better played on the Pi

Learn to program in Minecraft

How to code and hack with your favourite game

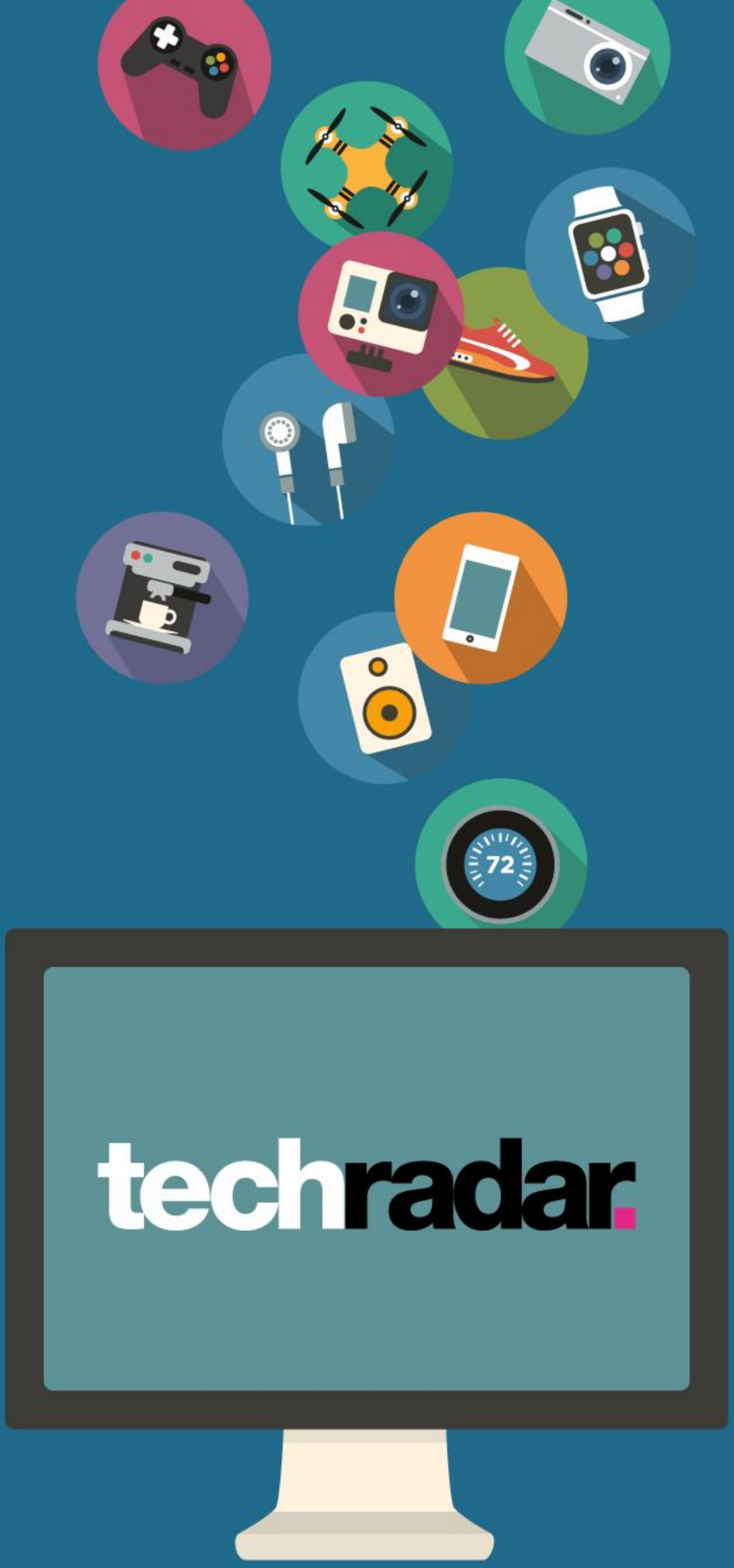
20 Awesome Pi add-ons

Make your Pi into something special

Future

PIU01 2016

SHOUT
SHARE
IT



The home of technology

techradar.com

PiUser

YOUR NEW SOURCE FOR RASPBERRY PI
JOIN THE PI REVOLUTION

Welcome!

If you've got a Raspberry Pi, or you're thinking of getting one, then you've come to the right place!

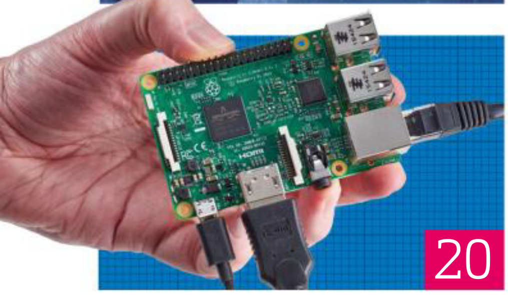
I grew up learning to code in Basic at home on my Acorn Electron. I typed in the code for games from magazines like *Electron User* and saved them on a cassette recorder. Eventually I got good enough to write my own games, and got one published in *Electron User* myself, and another one sold by a software company. This inspired me to pursue computer science all the way to University. Sadly, by then the home computers of my day had been replaced by Windows PCs, and programming by kids just for the fun of it was starting to become a forgotten art.

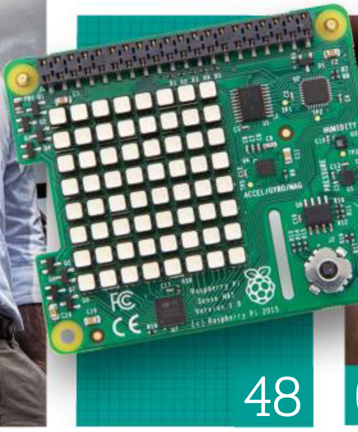
My love for computer magazines didn't go away, though, and ultimately that's why I'm here writing to you now. And thanks to the Raspberry Pi I feel like we've come full circle. The low-cost, revolutionary single board computer has brought back the programming bug among school children the world over, not to mention quite a few adults too!

At *Pi User* we're here to do more than get you started programming with the Pi, although we'll show you how to do that as well. We'll inspire you with the latest Pi projects, reviews and Pi news. Drop me a line and let me know what you think of our first issue. You can email me at **piuser@futurenet.com** and find us on Facebook at **www.facebook.com/PiUser/**

Graham Barlow Editor-in-Chief

Contents





WORLD OF Pi

Get your Pi connected and use the Pixels desktop

06 Eben Upton interview

10 Our favourite home Pi projects

14 Buying the right Pi for you

20 Get started with Raspberry Pi

22 How to install Raspbian

26 Using networks on the Pi

28 Master the Pixels Desktop

35 The top 20 Pi add-ons

Pi PROJECTS

Have an adventure with your Raspberry Pi

48 Get started with the SenseHAT

54 Recreate the Enigma machine

59 Build your own drone

64 Set up a cloud data server

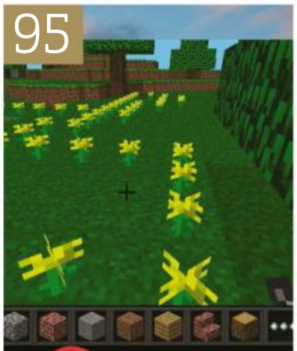
69 Build your first robot

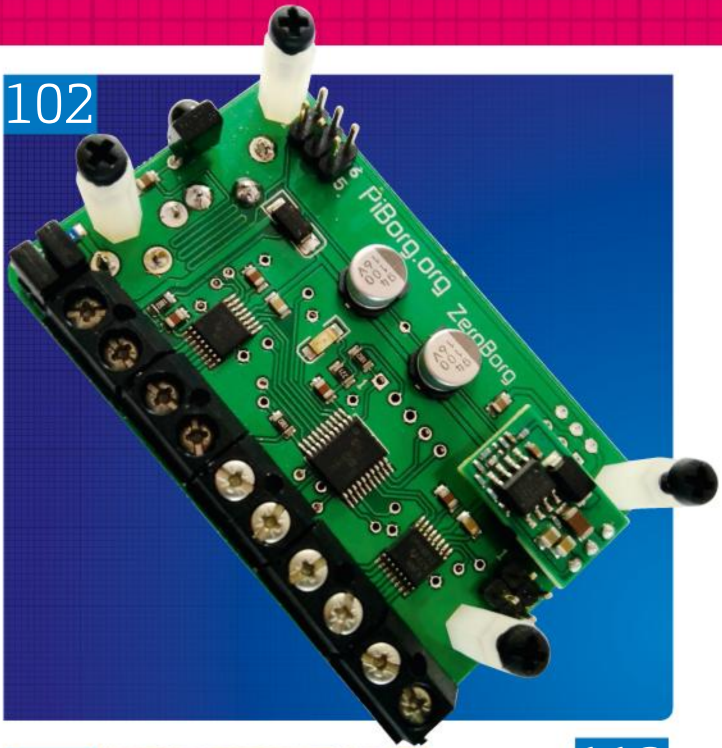
72 Master the Pi Camera Module

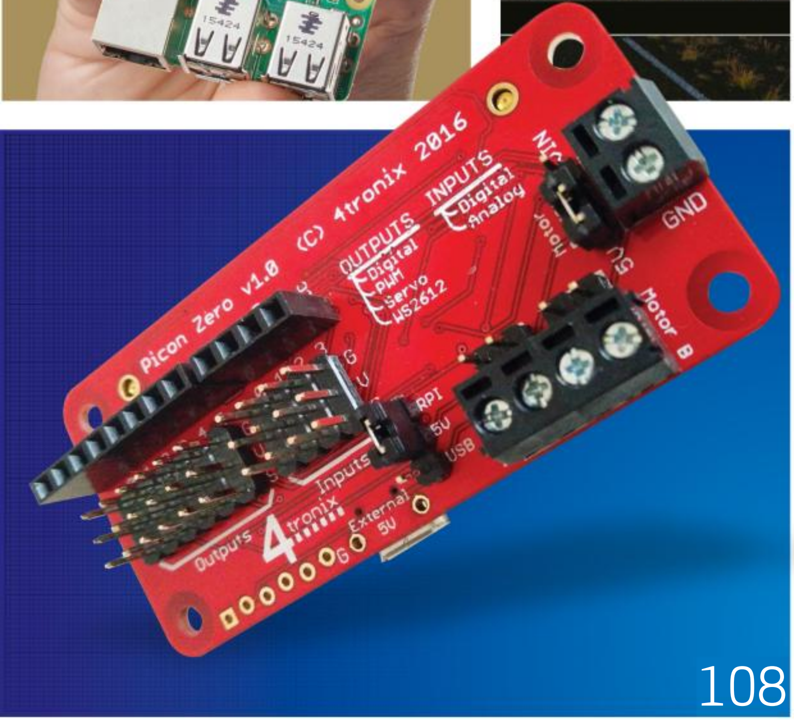
75 Make speakers Wi-Fi-enabled

78 Enjoy internet radio on the Pi

92


95


102


108

104

112

Pi CODE

Learn how to code in your language of choice

86 Get started with Scratch

92 Get to know Python coding

95 Begin coding in Minecraft

Pi REVIEWS

Tested and rated: the latest add-ons for your Pi

100 FUZE Workstation

102 PiBorg ZeroBorg

104 Drum HAT

106 Explorer HAT Pro

108 Picon Zero

110 CamJam EduKit 3

112 Picade by Pimoroni

Eben Upton

Pi User pays tribute to the inventor of the world's best-selling credit-card-sized computer, and finds out why the future of computing should still be firmly rooted in the 1980s…

Eben Upton, the Chief Executive of Raspberry Pi and a CBE (Commander of the Most Excellent Order of the British Empire), is the genius behind the Raspberry Pi and its success. He began tinkering with computers at an early age. When he wasn't sending off to BBC shows like *Think of a Number* for pamphlets about maths problems, he was copying reams of Basic code from computer books and magazines onto his trusty BBC Micro. Today, he's sold ten million Raspberry Pis and is CEO of a non-profit organisation that is dedicated to democratising computing.

To celebrate the first issue of *Pi User* we decided we needed to sit down with Eben Upton CBE to find out where it all started and what the future has in store for the Raspberry Pi Foundation, and to discuss the enormous contribution he's made to technology.

Pi User: **Were you always destined to work in technology? What do you think hooked you as a boy growing up in the 1980s?**
Eben Upton: I always had lots of type-in-code books, but I also had some higher-level, "So, what's a computer?" books. They were all telling me that in the future there would be a computer in every home – as common as having a telephone or cooker. It seemed incredible at the time.

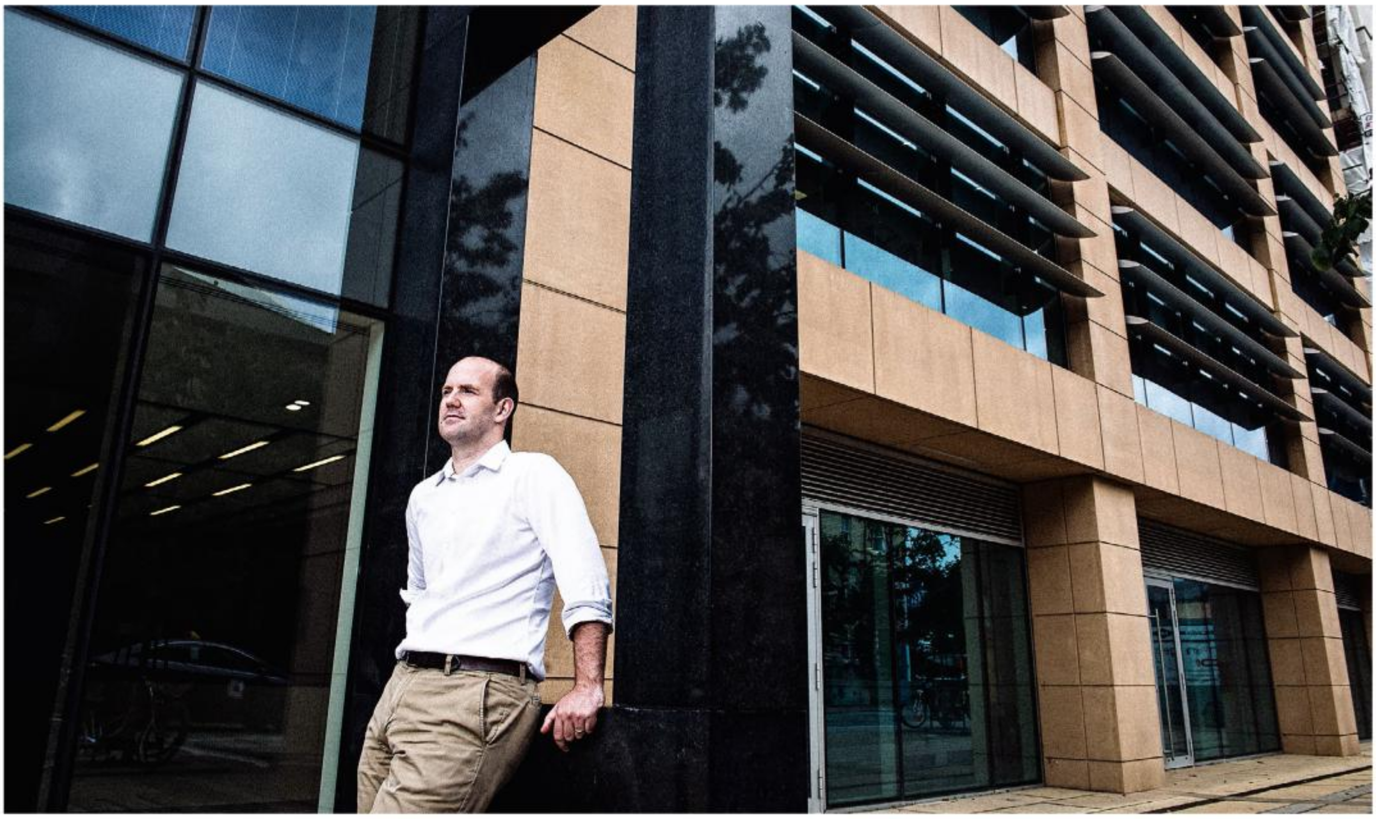
My youth made it really easy for me to be suckered into being a computer programmer. I think it helped that I'm one of those minority kinds of people who actually find computer programming itself interesting, rather than just the end goal.

Pi User: **What's the genesis behind the Raspberry Pi, beyond your obvious love for fruit-based tech-firm nomenclature?**
EU: I didn't really get into the hardware side of things until I finished my PhD in 2006. I was really burnt out and took a consulting gig with a guy who made motor controllers – he did a lot of *Robot Wars* hardware work before it became a popular TV show. This got me into AVR hacking, which is a really interesting halfway house between software and hardware. You write software that has this really low-level, deterministic interaction with the hardware, and that's what really got me interested in building computers, even considering my limited hardware skills at the time.

I built my very first computer back in 2006, and that was the very first thing you could call a Raspberry Pi. I then started working for Broadcom, but that idea of building a computer never went away. It wasn't until a bit later, when I was thinking about the computer-science recruitment crisis we had at Cambridge University, that the idea of making a small computer for the students to use came about. The notion of fixing this recruitment problem by making a piece of computer hardware seemed like a good idea at the time.

Pi User: **How has the reception of the Raspberry Pi differed from your original vision?**
EU: The success has been a real surprise. It's easily three or four orders of magnitude larger than we were ever expecting. We thought we'd sell between one and ten thousand units. Today, we're at ten million units.

Our big worry when we first started the Raspberry Pi project was that the kids wouldn't


2016 has been a fruitful (sorry) year for Eben and the Raspberry Pi.

care about it; that we'd make this thing and it just wouldn't work out. As far as we could see, there were two possible explanations as to why kids weren't programming any more: one was that they didn't have access to the hardware they needed to program, and the other was that they simply didn't care about it, that they just wanted to go on Facebook. It turns out that our first hypothesis was correct: the lack of a programmable platform was the only thing standing between a bunch of kids doing a bunch of programming.

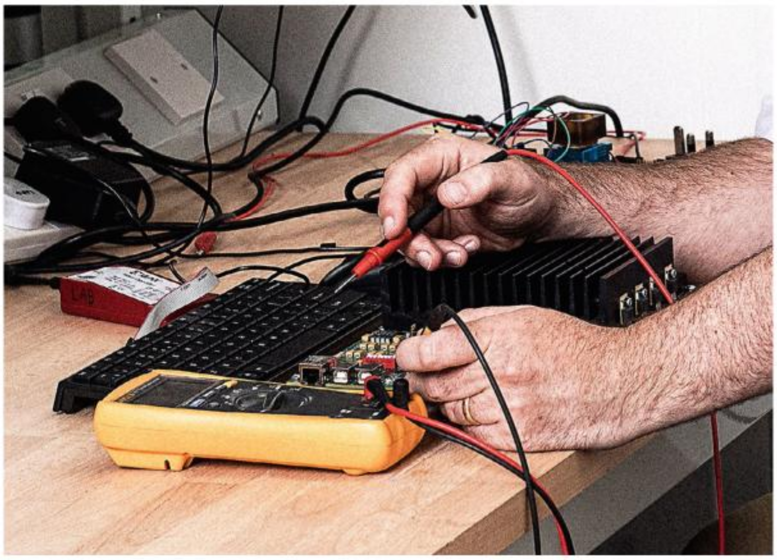
Pi User: Has your end goal changed as a result?
EU: Our goals now are basically the same, but much bigger. It's not just about an extra 200 applicants for Computer Science at Cambridge any more; it's about making sure no one in the world – including the developing world – is held back by the lack of accessible hardware. That's a satisfyingly vast goal. We're not yet anywhere near achieving that one, and we're not going to have achieved it until we've sold at least 100 million Raspberry Pis.

Pi User: Are you confident the Raspberry Pi is fulfilling the original goal of encouraging more kids to try computer science and follow careers in the technology sector?
EU: We're seeing an impact on the uptake of computer science in universities, and that's due to the combined efforts of a lot of organisations like the Raspberry Pi Foundation pushing in the same direction at the same time. Anecdotally speaking, I definitely think it's encouraging children into computing. All you've got to do is go to a community event like a Raspberry Jam, or look at what the kids are putting on Twitter and Facebook – there are thousands of kids doing things with Raspberry Pis, and it's a great sign for the future. Am I totally happy with the diversity? No. We're doing better, but we need to keep working to make sure we get girls as excited about computer science as boys. We need to continue to do more to help people from less advantaged backgrounds, too – that's a key area for us. A career in computing can be a great ladder for social mobility, and a great way for people to climb up into well-paid and rewarding careers.

Ultimately, though, I don't have a precise answer to that question, and that's a bad thing. As the Raspberry Pi Foundation scales up, we're continuing to do research… to go out and measure our impact and learn from it. We're finally in a position where we can do that.

Pi User: The Raspberry Pi Foundation teamed up with British astronaut Tim Peake, with the Astro Pi project. It was a great opportunity to raise public awareness. Did it work?
EU: I'm the kind of guy who's excited by computing, but most people aren't like that. Most need the eye candy, an extra interest factor to inspire them. Astro Pi has been a real success in generating excitement around computer science, and it's been successful in the UK. We've got two Raspberry Pis on the International Space Station and, provided we can get astronaut time, we'd love for it to continue – there's plenty of scope to take it further.

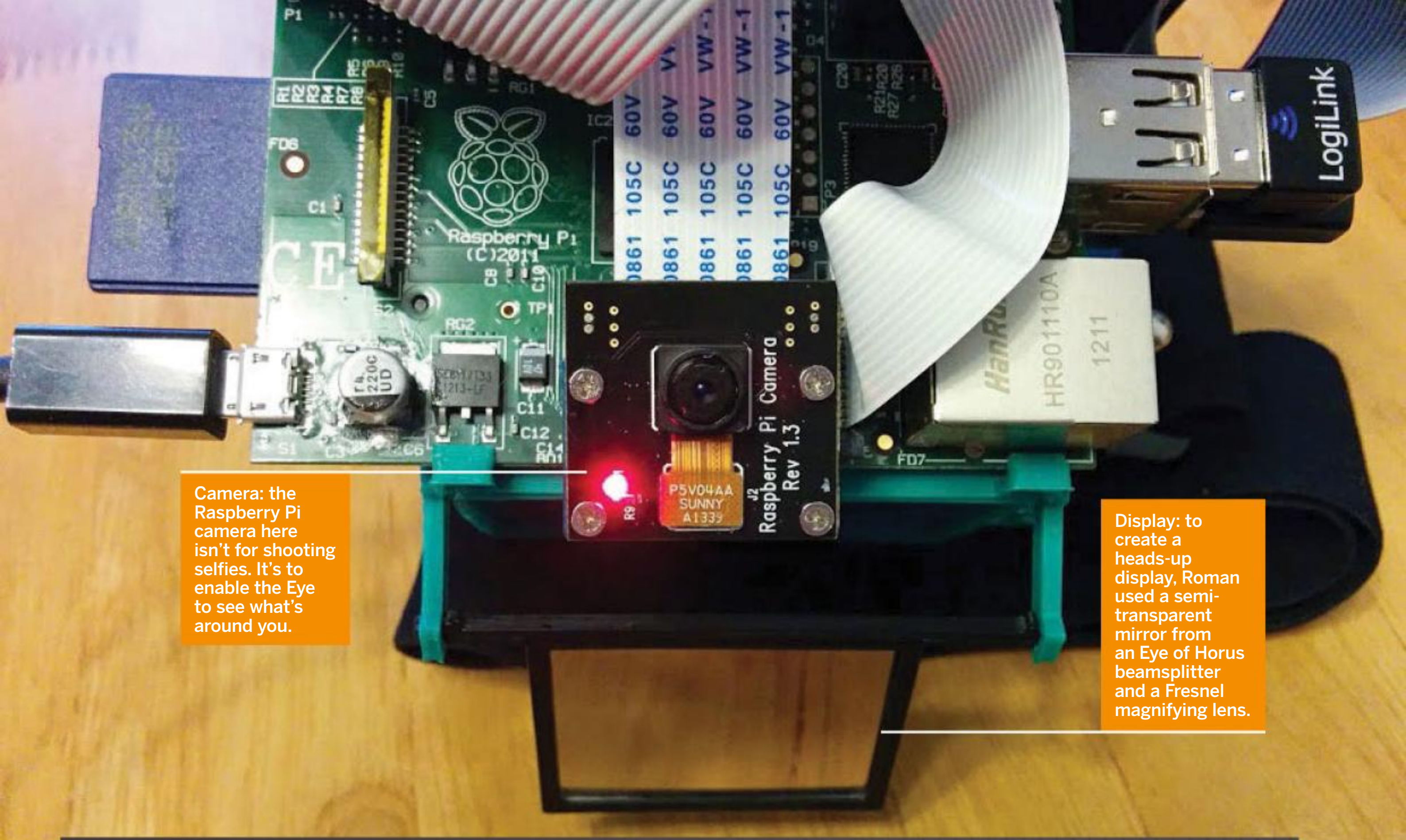
It was wonderful to see schoolchildren in the UK get so excited about Tim Peake going to space. When I was a kid, I used to care a lot about astronauts going to space, [but] do kids these days want to be astronauts? It turns out they do. It's great to see so many kids excited about technology in general, and not simply passive consumers of "stuff". Thankfully, they're as excited by the idea of someone climbing into a spacecraft with a million moving parts and going into space as I am.



Eben aims to inspire kids to "do it yourself" when it comes to technology.

Pi User: What sort of future do you envisage for hobbyist computing in general?
EU: There's always a risk of [it being] a flash in the pan. I think the key for it not to be a fad is to put proper infrastructure around it – teaching materials, teacher training, and things like after-school clubs, community events and so on. In the 1980s, there was a whole infrastructure around hobby computing, and we need to boot that back up in a stable way so that it can't wither and die again – it needs to be sustainable. There's much more involved with it than making a small computer. There's a lot of work to be done, but we're working on it and we're not alone in that. ■

> "THE SUCCESS [OF THE PI] HAS BEEN A REAL SURPRISE TO US. WE THOUGHT WE'D SELL BETWEEN ONE AND TEN THOUSAND UNITS. TODAY, WE'RE AT TEN MILLION UNITS"


Camera: the Raspberry Pi camera here isn't for shooting selfies. It's to enable the Eye to see what's around you.

Display: to create a heads-up display, Roman used a semi-transparent mirror from an Eye of Horus beamsplitter and a Fresnel magnifying lens.

Raspberry Eye

We love this head-mounted project that Pi enthusiast Roman Rolinsky has created in his spare time

RECIPE

- ITDBo 202.4E LCD screen
- 1 x Eye of Horus beamsplitter
- 1 x Fresnel magnifying lens

Some projects are created to solve a problem. Others are created out of pure curiosity, a bit of ambition and a big sense of fun. Roman Rolinsky's Raspberry Eye head-mounted display falls into the second category: while it's an impressive bit of engineering, Roman admits "it was simply too bulky for something practical. I've started to work on a smaller version using a 1-inch display paired to a Wi-Fi transmitter, but optics need to be much more advanced to create a see-through image at a comfortable distance," he says. "Maybe printing custom optical elements is possible with high-end 3D printers – this is a direction I will try to explore." Roman is excited about the possibilities of 3D printing, and has pre-ordered an OLO 3D printer via its Kickstarter page.

Was the Eye his first bright Pi idea, or his last? "Both, actually," Roman smiles. "The Pi used for the Eye project soon died, because the board is not really designed to be worn on a headstrap without any protection. I've got the second generation after that, but I haven't used it for anything DIY yet."

Like many builders, Roman has shared the full details of his project online. How important is the community aspect of Pi building to him? "It was really important to have notro's FBTFT kernel driver as an open-source project with good documentation," he says. "I just had to adapt some code for the display I was using and to produce a mirror image. From the comments I've received on Hackaday I saw there are people interested in collaborating on the project, so I will probably contact some of them if the next iteration finally becomes active. And of course, it is very nice to see the stuff you are doing noticed by the maker community."

So what other projects does Roman have his, ahem, Eye on? "I've bought my first house recently," he says. "It's pretty old and requires a lot of attention, so it's probable that my next project will be using Pi for home automation."

› The Pi isn't really designed for head mounting, so the Eye is a little on the large side.

CREATOR

ROMAN ROLINSKY

Roman is a self-employed software developer, 3D printing enthusiast and open source advocate.

Magic Mirror

Mirror, mirror on the wall, what is the smartest Pi project of all? Pi fans say this is number one

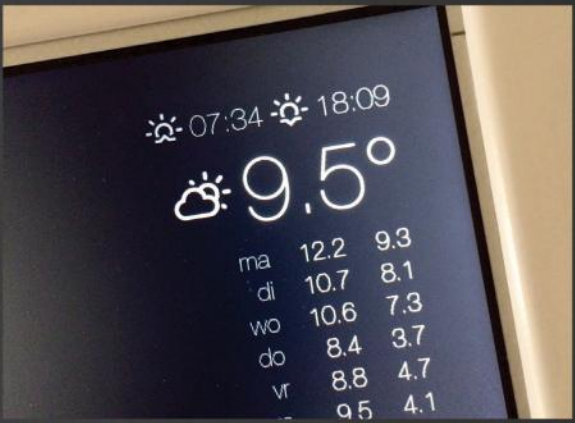
Michael Teeuw's Magic Mirror is the kind of thing you wish you'd thought of: a mirror that projects information for you while you make yourself presentable. It was voted best Pi project of all time by the Pi Foundation's official magazine, and deservedly so: not only is it really clever, but it has inspired countless builders to make their own. "I've seen many, many cool use cases for the Magic Mirror, but my absolute favourite is one of the community members that uses the Magic Mirror project to teach kids the possibilities of programming by letting them modify and create their own modules," he says. "If the Magic Mirror project could inspire one kid to become a programmer, I would be absolutely thrilled!" It's likely to inspire more: Michael has shared his code and built an entire platform for mirror makers at **https://magicmirror.builders**, and there are several possible commercial spin-offs on the horizon.

Michael had the idea during a shopping trip, when he stared at an illuminated mirror and imagined it providing him with useful data as well as reflecting his own bored expression. "Wandering around behind your shopping girlfriend is a great moment to think of new possibilities," he laughs. The mirror shows the time, weather forecast, news and a nice compliment to cheer you up, and he's published the entire project details at **http://michaelteeuw.nl/tagged/magicmirror**. So has he had any other retail-related revelations? Sadly not – but "walking through a big city, in this case New York, can be a major source of inspiration."

For Michael, Pi building is "a great way to improve your own programming skills and help others to improve theirs... If it's out in the open, you want to deliver a quality product. I also find it fascinating that others have built so many different modules for the Magic Mirror platform, modules I would never have thought of. That shows how we can all benefit from the enthusiastic and rapidly growing building community."

RECIPE

- **1 x Iiyama 24in monitor**
- **1 x sheet of one way mirror glass, alias observation mirror glass**
- **1 x HDMI cable**



The mirror's possibilities are endless: if it can be displayed on the screen, it'll appear through the glass.

Good morning, handsome!

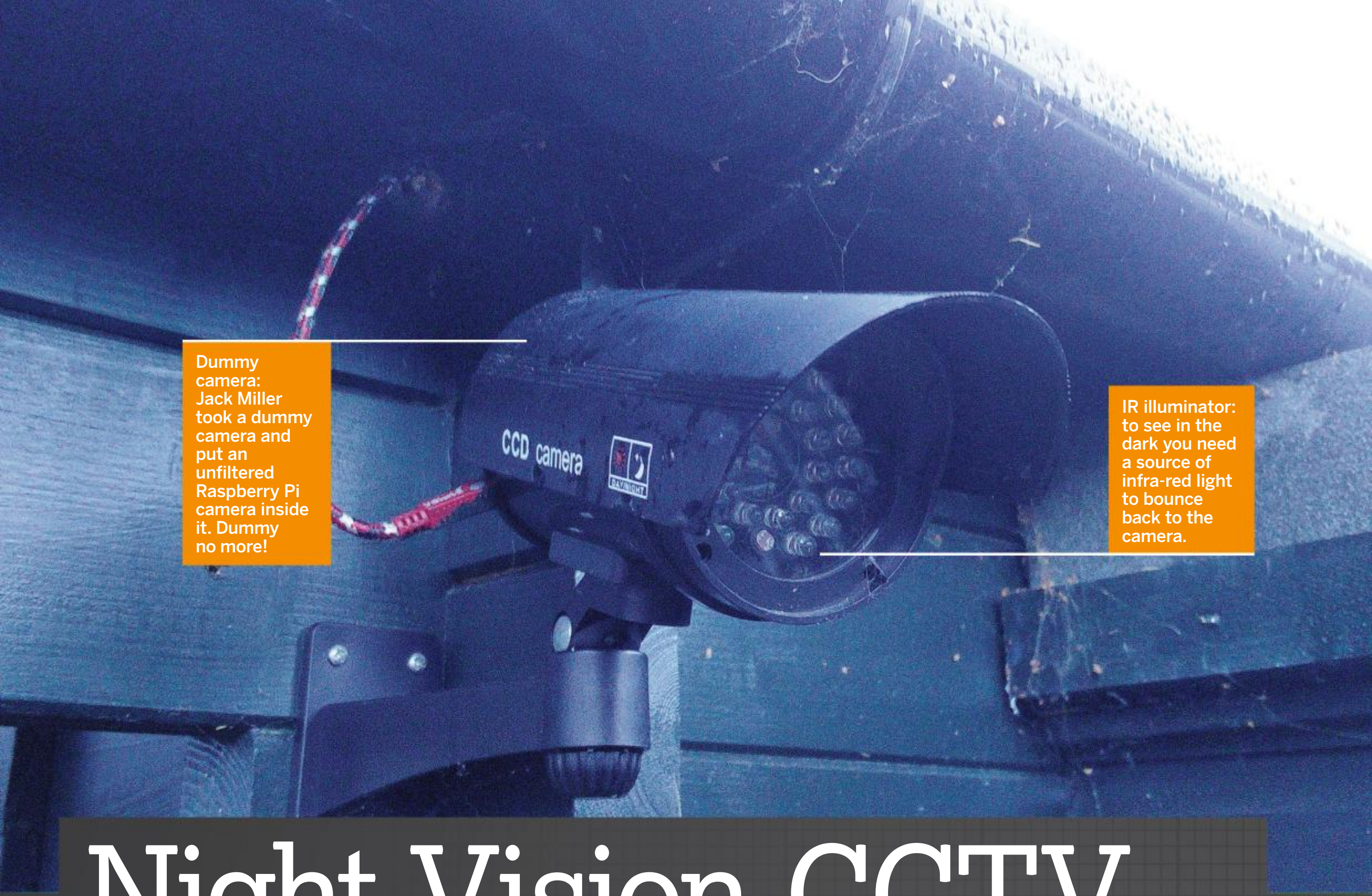
Wooden frame: this is probably the trickiest bit; the tech is really straightforward, but putting the glass in the frame is a bit fiddly.

TFT monitor: the Iiyama monitor is nice and thin, leaving enough room to mount the PI and wiring without making the frame too big.

CREATOR

MICHAEL TEEUW

Michael likes to make things, both physical and virtual, and blogs about them at **http://michaelteeuw.nl**



Dummy camera: Jack Miller took a dummy camera and put an unfiltered Raspberry Pi camera inside it. Dummy no more!

IR illuminator: to see in the dark you need a source of infra-red light to bounce back to the camera.

Night Vision CCTV

Keep an automated eye out for burglars, Santa and other unexpected visitors with this clever CCTV setup

RECIPE

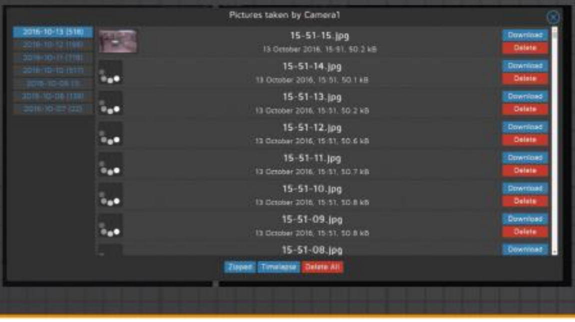
- ● **1 x Raspberry Pi camera without IR filter**
- ● **1 x dummy CCTV camera**
- ● **1 x IR illuminator**

When Jack Miller moved out of his house and into a cabin in his garden – "weird, I know," he says – he wanted to be able to monitor his house in case of break-ins. So, he created a Raspberry Pi CCTV camera with night vision. He also found that in the daytime the camera had another benefit: Jack makes popular video tutorials (look up Jackktutorials on YouTube), and the camera would let him see whether anybody was about to interrupt him while recording. As for the night vision, that "was mainly a feature that I thought was cool, and the no-IR filter camera was cheaper when I actually came to buying the parts. It just seemed cool, I guess."

The video Jack made of the project makes the entire process look effortless, but we got Jack to admit that some moments did end up on the cutting room floor. "I did break three CCTV dummy housings trying to solder the IR LEDs onto them, which I eventually gave up on, and of course I burnt myself numerous times," he says. "Then I dropped the ball joint down a space in my decking outside, so I had to buy another one. These incidents are better left out of the video."

Jack is a big fan of the sharing ethos around the Pi. "Sharing and teaching is what I do best, and I've been doing it for a long time," he says. "I got this particular idea for the camera from a guy on YouTube called Clayton Lambert, who demonstrated a motion detection script. I then found ccrisan, who developed motioneyeOS, and thus the night vision camera was born."

What's next for Jack? "I've just finished a Pentesting Dropbox using the Raspberry Pi," he told us. "It's a small device with Kali Linux installed. You can drop it into a network and be able to access it and conduct pentesting. That was pretty fun and complicated, but I'm definitely going to be using it in more videos. I also have an idea for the Pi becoming an SSH server, and I think I'm going to rebuild my home automation system." Jack also plans to resurrect a broken Nintendo Entertainment System with a Pi inside. Watch this space!



15-51-15.jpg
13 October 2016, 15:51, 50.2 kB
15-51-14.jpg
13 October 2016, 15:51, 50.1 kB
15-51-13.jpg
13 October 2016, 15:51, 50.2 kB
15-51-12.jpg
13 October 2016, 15:51, 50.6 kB
15-51-11.jpg
13 October 2016, 15:51, 50.7 kB
15-51-10.jpg
13 October 2016, 15:51, 50.8 kB
15-51-09.jpg
13 October 2016, 15:51, 50.8 kB
15-51-08.jpg

› The camera can deliver a real-time video feed as well as capturing still images at preset intervals.

CREATOR

JACK MILLER

Jack's a busy man. The **Jackktutorials.com** site features reviews, forums and a popular YouTube channel.

Game Boy Zero

Could this be the best Game Boy ever made? Meet the man who used a Raspberry Pi to make the perfect Game Boy

Warner Skoch says "The original Game Boy has been one of my favourite consoles since I was a kid. I've also been really interested in electronics and tinkering in general, so when I saw the Raspberry Pi Zero's capabilities, this was one of the first things I wanted to try."

The resulting project "blew up" – but not in a Samsung Galaxy Note sense. When Warner posted details of his Game Boy Zero on Reddit, the story made the pages of *The Verge* and *Popular Mechanics*, among many others. "I've had hundreds of people asking to buy one from me, hundreds more asking for a how-to guide, and one marriage proposal." Making more isn't really practical for Warner, so he decided to post full guidance on **Sudomod.com** and hang around to help others with their own Game Boy projects.

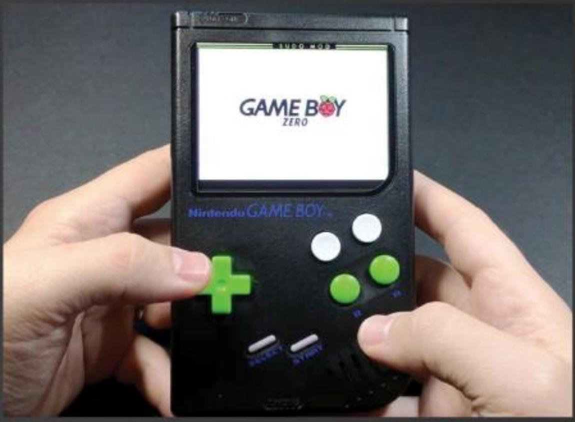
You can understand the excitement, because Warner, alias Wermy, has done some really cool things. He's adapted a Game Boy cartridge to work as a microSD card reader, and that means the Game Boy Zero can play games from it. He's also added extra buttons to play other Nintendo games including NES and SNES. And he's a big fan of video editing to hide the fact that "there was plenty of swearing and even some bleeding." For him, the SD card reader was the worst bit. "It was the most headache-inducing, given how finicky SD cards can be."

For Warner, the builder community is "the best part" of Pi projects. "I recently did a second guide using all-custom and aftermarket parts like custom button PCBs, many of which the community – and in particular the folks on the Sudomod forums – came up with specifically for this project. It has just been so much fun to be part of."

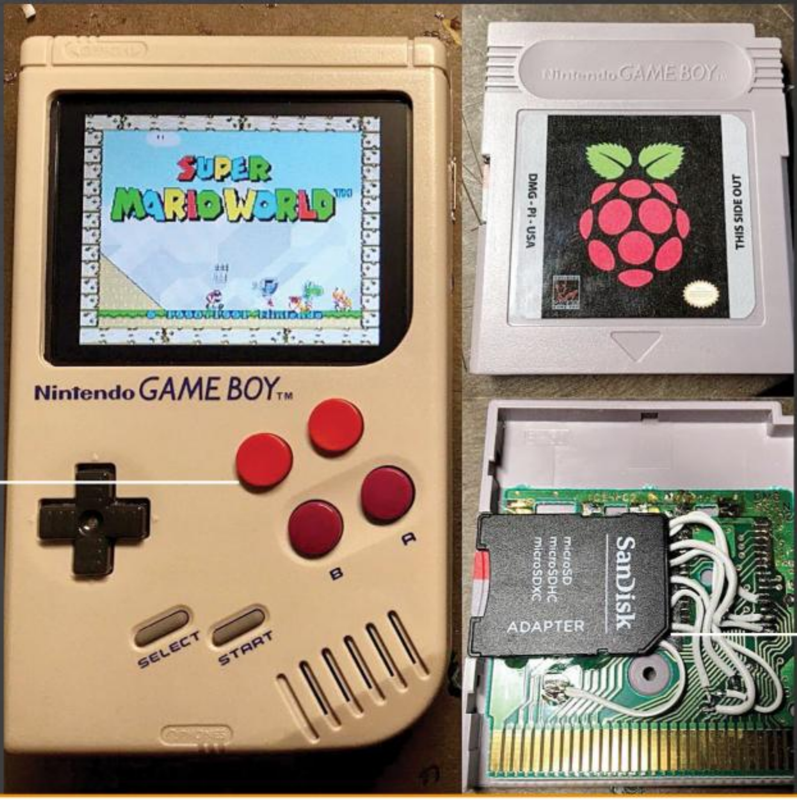
We love Warner's quirky ideas – his most recent project was a Duck Hunt gun modified to be a lamp zapper, with the IR receiver hidden behind a duck. What'll he come up with next? "I've got a few things on my to-make list," he says. "Stay tuned!"

RECIPE

- **1 x broken Nintendo Game Boy**
- **1 x Adafruit 3.5in composite display**
- **1 x Adafruit 2500mAh battery**



The Game Boy Zero supercharges mobile gaming with Pi power.



Extra buttons: Wermy added two more buttons so that the Game Boy Zero could play NES and SNES games too.

SD card reader: the card reader is cunningly housed in a Game Boy cartridge to keep things retro while offering stacks of storage.

CREATOR

WARNER SKOCH

Warner, alias Wermy, posts his projects and receives marriage proposals at **sudomod.com**.

What Pi to buy

Familiarise yourself with the types of Raspberry Pi and accessories on offer

The Raspberry Pi is what is known as a "single board" computer. In plain English, this is a complete computer built into a single circuit board. Single board computers like the Pi offer many advantages over ordinary computers. First, they are a good deal smaller – the Raspberry Pi is only about the size of a credit card. Second, the Pi draws very little power (around 5V), so can run indefinitely without any risk of overheating and does not require a noisy, bulky fan. It's also very inexpensive to mass produce Raspberry Pis, hence the low cost of purchasing one.

One of the down sides to having an entire computer on a single circuit board is that it's difficult to upgrade individual components such as the RAM (memory). At the same time, the Raspberry Pi is not as simple to use out-of-the-box as a new smartphone or tablet computer might be – you need to know how to set it up, where to plug in the video, the storage, and so on. The low cost is also sometimes a little misleading because you need to add extras such as cables for power and a monitor, not to mention the monitor and keyboard themselves, in order to do anything with your Pi.

If you've just bought your first Raspberry Pi, congratulations! In the pages that follow, you'll find out all about how to set it up and what it can do. If you don't yet have a Pi and simply want to know what to buy to get started with Raspberry Pi, then we'd recommend purchasing a Raspberry Pi 3 Starter Kit. These are available from vendors such as the Pi Hut (**thepihut.com/products/raspberry-pi-3-starter-kit**). Although you should check carefully to see what's bundled with the Pi from various vendors, starter kits usually include the Raspberry Pi 3 Model B itself, an SD card with NOOBS pre-installed (more on NOOBS later), a power supply, a case and usually an HDMI or network cable.

Not only will the Raspberry Pi 3 be suitable for any of the projects in these pages but it is fully backwards compatible, so will work for any projects you may find for earlier Pi models.

The Raspberry Pi 3 is most useful for resource-heavy projects like playing games or streaming video. If you don't need to do this, then there are less costly Pi models available.

A brief history of Pi

The very first generation of Raspberry Pi (known, a little counter-intuitively, as the model B) was released in 2012. This was replaced in July 2014 by the Model B+, which introduced various technical improvements and several changes: among other things, the original Pi had two USB ports and 26 GPIO (General Purpose Input/Output) pins, but the Model B+ had four USB ports and 40 GPIO pins.

In February 2015, the Raspberry Pi 1 Model B+ was superseded by the Pi 2 Model B. This was in turn replaced in February 2016 by the Raspberry Pi 3, the third generation Raspberry Pi. Apart from being more powerful, the main difference is that the Pi 3 has built-in Wi-Fi and Bluetooth.

In 2013, a low-cost variant of the Raspberry Pi, the Model A, was introduced. This was replaced by the Model A+ in November 2014. The main difference between the Model A+ and the Pi 3 is that the Model A+ does not have Ethernet, Wi-Fi or multiple USB ports. It is really intended for embedded projects – that is, building into devices that don't require connectivity.

It's not necessary to know every model of Raspberry Pi, but if you read older magazines or browse websites for Pi projects, they might refer to these models being required for projects. As we've noted, the Pi 3 is fully backwards compatible, meaning it will work for any such projects.

Some websites do still sell older models, and you might be tempted by these because they are a little cheaper than the Pi 3.

THE RASPBERRY PI 3 IS FULLY BACKWARDS COMPATIBLE, SO IT WILL WORK FOR ANY PROJECTS YOU FIND FOR EARLIER MODELS

Pi bites

If you have an older version of the Raspberry Pi and want to use wireless, there is an official Raspberry Pi wireless adapter, available from sites such as the Pi Hut (**http://bit.ly/PiWiFi**).

❮ Sites like the Pi Hut sell a starter kit which includes the Pi itself, as well as the necessary cables, SD card and a case.

This is your decision to make, but bear in mind that the older models are not as fast, consume slightly more power, and lack built-in Bluetooth or Wi-Fi. If a project requires very low power draw, consider the Model A+ or Pi Zero (more on this shortly).

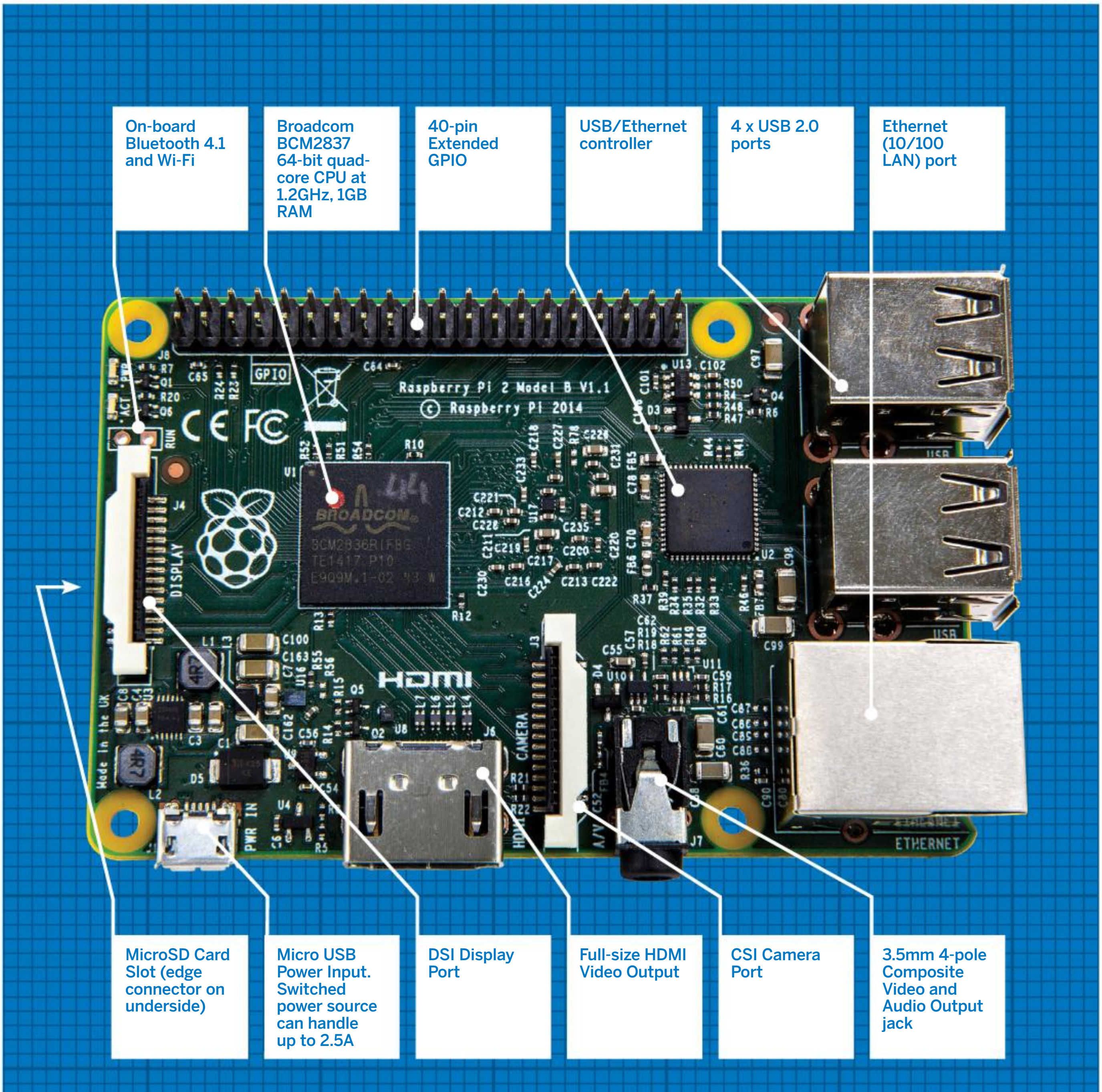
Raspbian & NOOBS

While looking at the various Raspberry Pi models, you're likely to see references to both Raspbian and NOOBS. Raspbian is a free operating system (OS) designed to work specifically with the Raspberry Pi. It's based on Debian Linux, which is an OS for desktop computers. It contains a number of handy pre-installed programs and utilities, enabling the Pi to function out-of-the-box.

Raspbian is in fact only one of a number of operating systems that will work with your Pi, but there's rarely any compelling reason to try to install a different OS unless it's specifically recommended for the project you're following.

If you buy a Raspberry Pi Starter Kit, it's very likely that Raspbian will have been installed already onto your microSD card through NOOBS (New Out Of the Box Software).

NOOBS is simply an operating system installer which allows »


On-board Bluetooth 4.1 and Wi-Fi
Broadcom BCM2837 64-bit quad-core CPU at 1.2GHz, 1GB RAM
40-pin Extended GPIO
USB/Ethernet controller
4 x USB 2.0 ports
Ethernet (10/100 LAN) port
MicroSD Card Slot (edge connector on underside)
Micro USB Power Input. Switched power source can handle up to 2.5A
DSI Display Port
Full-size HDMI Video Output
CSI Camera Port
3.5mm 4-pole Composite Video and Audio Output jack

The Raspberry Pi 3 dimensions are a compact 85.6mm x 56mm x 21mm.

you to select the OS you want to install from a friendly menu. This is not only convenient if you want to try a different OS; it is extremely useful if you decide to start a new project on your Pi as it enables you to reset the Pi to how it was when you first took it out of the box. Simply restart the Pi, then hold down Shift on your keyboard and you can opt to install Raspbian afresh.

If you've got a brand new Raspberry Pi, you should also make sure you have the most up-to-date version of Raspbian installed before starting any project. You can do this by opening the Terminal app and typing the following commands:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

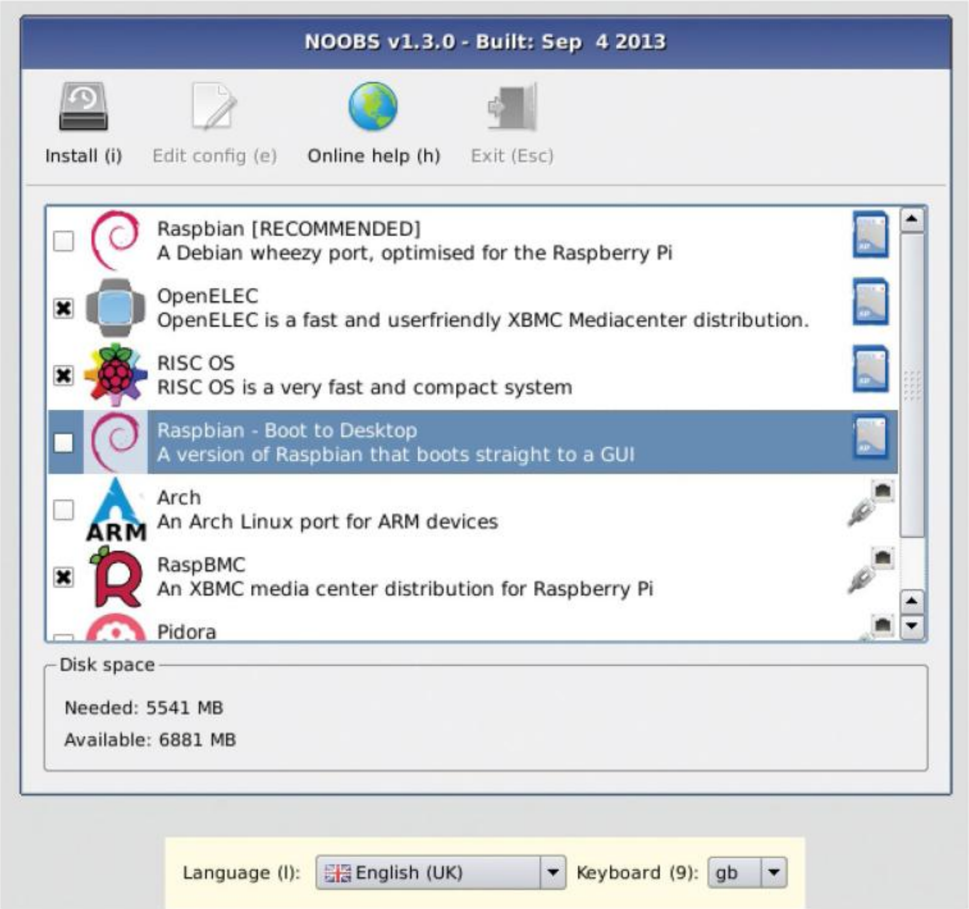
Press Return after entering each command.

The Raspberry Pi 3

As we've noted, the Raspberry Pi 3 is the third generation of the Pi. Unlike its predecessor, the Raspberry Pi 2, it supports both Wi-Fi and Bluetooth connection out of the box without the need for separate adaptors.

If you're familiar with computer hardware, it has a 1.2GHZ 64-bit Quad-Core processor and 1GB of RAM.

The Pi 3 also has four USB ports. While most USB devices will work, some will require more power than the Pi can supply. If you want to use the Pi with a



NOOBS v1.3.0 - Built: Sep 4 2013

☐ Raspbian [RECOMMENDED]
A Debian wheezy port, optimised for the Raspberry Pi

☒ OpenELEC
OpenELEC is a fast and userfriendly XBMC Mediacenter distribution.

☒ RISC OS
RISC OS is a very fast and compact system

☐ Raspbian - Boot to Desktop
A version of Raspbian that boots straight to a GUI

☐ Arch
An Arch Linux port for ARM devices

☒ RaspBMC
An XBMC media center distribution for Raspberry Pi

Pidora

Disk space
Needed: 5541 MB
Available: 6881 MB

Language (l): English (UK)
Keyboard (9): gb

The NOOBS installer gives you a choice of OS to install on the Pi. Select Raspbian again to restore your Pi to factory settings.

more power-hungry device such as an external hard drive or HD webcam, consider buying a powered USB hub, such as Pi Hut's 7-port powered USB hub (**thepihut.com/products/7-port-usb-hub-for-the-raspberry-pi**). A powered hub connects both to the Pi and a mains plug socket to provide the required juice, so will be able to power any USB device.

If you can't or don't wish to use wireless, the Pi 3 does have an Ethernet port for networking using a cable. If you haven't purchased a starter kit that includes one, make sure you also get an Ethernet cable. The Pi 3 also has an HDMI port which can be connected to any HDMI monitor or television. There's also an analogue video/audio port, as well as a port to connect the official Raspberry Pi Camera.

The Raspberry Pi 3 has 40 GPIO (General Purpose Input/Output) pins. You'll find these along the edge of the board. These are used for hardware projects to do things like push buttons, light LEDs and connect sensors. (Some Pi starter kits come with a few breadboards and LEDs to begin your foray into physical computing.) For more information on the GPIO pins visit **www.raspberrypi.org/documentation/usage/gpio/**.

If you aren't planning to attempt any hardware projects, feel free to ignore these, but it's a good idea to carefully examine each side of your Pi to identify each port.

Unlike smartphones, tablets and desktop computers, the Pi does not have data storage built-in. This needs to be added in the form of a microSD card, which slides into a port on the end of your Pi. Make sure the Pi is powered off before removing it. While on the subject of power, the Raspberry Pi doesn't have a built-in power switch. You can however shut it down from within Raspbian.

Raspberry Pi Zero

The Raspberry Pi Zero is both smaller and far less power-hungry than the Raspberry Pi 3. It's designed for projects where size is critical and/or where you don't need all the features of the Pi 3 such as Wi-Fi and Bluetooth.

The Pi Zero is also a good deal cheaper than the Pi 3 – as little as £4 from the Pi Hut. That said, the miniaturisation comes at the cost of being incompatible with regular sized devices unless you use a special adapter.

For instance, the Pi Zero doesn't have a regular-sized USB port, but it is possible to buy a micro USB adapter to connect normal USB devices. Similarly, regular sized HDMI cables cannot be connected to the Zero without a special mini HDMI adapter.

Aware of these shortcomings, the Pi Hut website has released an "Essential Raspberry Pi Zero Kit" which includes not only the above adapters but also some small rubber feet, some GPIO adaptors and a handy metal tin to store everything in. See **thepihut.com/collections/raspberry-pi-zero/products/raspberry-pi-zero-essential-kit** for more information. »

It's easy to forget how small the Raspberry Pi board is. Even the Raspberry Pi 3 feels tiny in your hand.

Zero
Essentials Kit
The PiHut
Power Adaptor
ThePiHut
SAMSUNG
8
HDMI
GPIO
MICRO SD CARD

› Raspberry Pi Zero as sold by the Pi Hut. Above are a power adapter and the "Essentials Kit". To the left is a micro SD, USB wireless adapter and shim.

If you need internet access on the Pi Zero, then the Official Raspberry Pi Wireless Adapter will work in conjunction with the Zero's mini USB connector. Alternatively, Pi Hut sells a MicroUSB converter "shim" which slots over the end of any USB device, saving trailing cables. Visit **thepihut.com/products/usb-to-microusb-otg-converter-shim** to find out more.

The smaller size and price of the Zero comes at the cost of performance. It has a 1GHZ single-core processor and just 512MB of RAM. However, this is actually ample for many Pi related projects.

Raspberry Pi Compute

Finally, the Raspberry Pi Compute Module contains the bare bones of the Raspberry Pi (a processor and 1GB of RAM) but instead of using an SD card it contains 4GB of internal flash memory. The Compute Module is designed for specialised industrial applications, so it's unlikely that you'll need to use this, but we've mentioned it here in case you see it for sale.

The small size of the Compute Module (67.6x30mm) makes it ideal for a number of ambitious projects such as the CubeSat project, which hopes to build shoe-box-sized satellites that can be launched into space, all powered by the Pi. ■

SAVE UP TO 58%
FROM JUST £5

A GREAT GIFT FOR EVERYONE

Treat yourself or a loved one this year with a magazine subscription

LINUX FORMAT
Windows Help & Advice
Mac FORMAT
macOS Sierra
iPad User Magazine
Meet iOS 10
T3
101 GADGETS YOU CAN'T LIVE WITHOUT

- Choose from a huge range of titles
- Save up to 58% off the cover price
- Subscribe from just £5 every 3 months
- Free personalised gift card when buying for someone else

2 EASY WAYS TO ORDER

VISIT myfavouritemagazines.co.uk/Z704

OR CALL US ON 0344 848 2852

Please quote Z704 when ordering by phone.

Lines are open Monday to Friday 8am to 7pm and Saturday 10am to 2pm (GMT).

Savings calculated against the full RRP (single issue price x frequency). This offer is for new subscribers only. Full details of the Direct Debit guarantee are available upon request. You can write to us or call us to cancel your subscription within 14 days of purchase. Your subscription is for the minimum term specified and will expire at the end of the current term. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at point of print and subject to change. For full terms and conditions please visit: myfavm.ag/magterms. Offer ends: 31st January 2017.

Future myfavouritemagazines
Official Magazine Subscription Store

Get started with a Raspberry Pi

Before you can actually do anything with your Raspberry Pi, you're going to need to hook up all the right peripherals. Here's how...

Even though it might not look like it, the Raspberry Pi is in a fundamental way just like any other desktop computer you might have owned or used in the past. That is to say, it has a processor, memory and storage and, in order for you to use it, it requires being connected to all the usual peripherals – a display, inputs and more. For most people, this should be straightforward, but there are some finer points that you should know about – details about the correct SD card to use and compatible wireless adaptors, for example – before we can move on to creating a boot disc to start it up.

To use the Pi as a computer, you'll need to connect at least a keyboard, a mouse, a display and a compatible SD card. In today's environment, it's going to be useful to at least have a wired internet connection around, or a suitable USB wireless adaptor for wireless networking. The final touch would be desktop speakers. On a technical level, peripheral support is provided by the Linux kernel that powers the Raspbian OS, but more on that later...

Need input!

Ever since the Model B+, the Raspberry Pi has been well furnished with four USB ports. This ensures that it's easy to connect all the peripherals you need. The base minimum of these is a USB mouse and keyboard for standard input – especially considering you're able to get keyboards that provide additional USB ports, helping to save one or more on the Raspberry Pi itself.

Pretty much any keyboard and mouse should work, but don't expect support for any unusual buttons (you'd be surprised at how many do work, though...).

The important thing to remember is that even with just four USB ports – or the single port on the Pi Zero – it's easy to expand your USB connectivity with a hub. For devices that don't require additional power, an unpowered hub is an option, while powered hubs can handle any devices, no matter whether power is required or not.

Display me

Technically, the Raspberry Pi works with most types of displays using the correct cable or adaptor. It's envisioned to be used with an HDMI monitor but works with older monitors with a DVI adaptor or an HD TV. With the HDMI connector, both the Pi and the display should automatically configure themselves correctly; we've successfully used the Raspberry Pi on a 4K display without issue, beyond manually setting the higher 4K screen resolution.

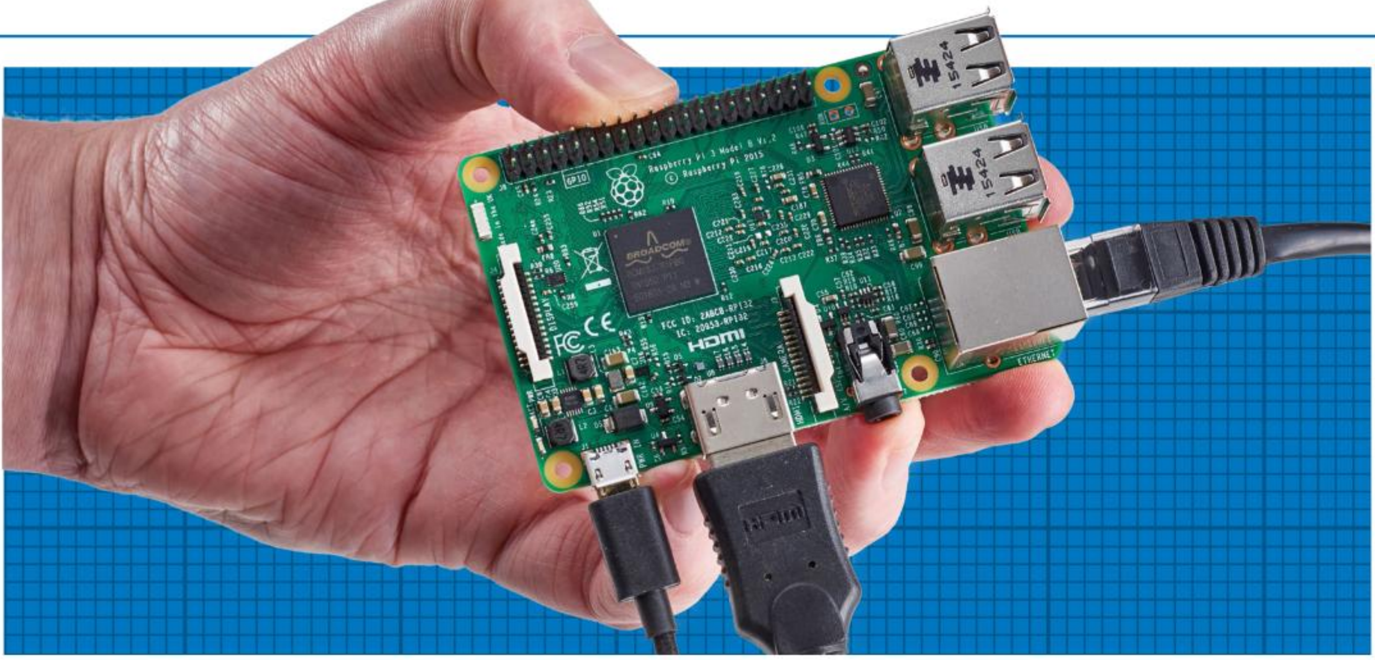
It's possible to connect the Pi to an old TV or monitor using an analogue composite connector – this is usually a round yellow RCA connector. With the Pi, the 3.5mm jack that's usually used for stereo audio output carries an extra line for the video. To access this, you need a 3.5mm jack to stereo and composite RCA.

Sounds good

The Raspberry Pi is perfectly capable of producing plenty of audio, and later we'll see projects that enable the use of the Pi as an audio streamer and media centre. You can get audio out of standard Pis in two ways. The first is over the HDMI line. When you plug the Raspberry Pi into a display that also has speakers, audio generated by the Pi is automatically sent over the HDMI line, too.

ON THE CASE

We admit, it can be unnerving having a naked Raspberry Pi sitting on your desktop, and it's certainly the case that it could short on random wires, screws or metal surfaces. So, a common extra that people get for their Pi is a suitable case. At this point, there's quite a range available, and many leave the GPIO pins and other ports open for easy access. There are more exotic options that can mount the Pi within a display, a desktop-style case that expands its capabilities, and even a laptop chassis, so you can take your Pi with you. Just be aware that this means you can spend anything from £5 up to £100 for your Pi case!



Connect it up and a Raspberry Pi will form the heart of a powerful but very compact computer setup.

The alternative option is to use the standard 3.5mm audio jack, which can easily be plugged into most speakers.

Network and power

We're going to look into Pi networking in more detail over the page, largely because we want to look at wireless networking adaptors and the newer Raspberry Pi 3 wireless capabilities in full, alongside the new Bluetooth features and how they work. Beside this, all Model B Pis come with the wired Ethernet network port. This can be plugged into any router, and boom – instant network access, including internet access if that router is connected. For the Model A Pis and Pi Zero, a USB wireless adaptor is required.

All Raspberry Pis use a micro USB port to supply their power. This has the advantage that, on the whole, you can power a Pi from almost any phone power adaptor. The power requirements vary, depending on which Pi you have and what you're doing with it. Old power supplies that provide only 5W should cope with a basic Pi Model B setup, which draws 3.5W, but the more peripherals you add, the more power it requires. The Pi 2 and 3 use 4W, so the draw is more critical. Ideally, you want a 10W (2A, 5V) supply, which is more common these days because modern tablets and phones also have higher power requirements.

At this point, you would have your Raspberry Pi all set to be a normal desktop PC. The fact is, the Pi was envisioned to be that and so much more. A big part of its extra abilities comes through the bank of GPIO (General Purpose Input Output) pins. We'll explore these later through fun and exciting projects, but for now, it's enough to know that these enable the Pi to control, monitor and power external devices and projects. You don't have to use them, but they're always there if you want them. Additionally, the Pi comes with a camera interface – although you can also use standard USB cameras – plus there's a dedicated digital touch display interface, too.

All you have to do now is follow our installation guide over the page and boot it up!

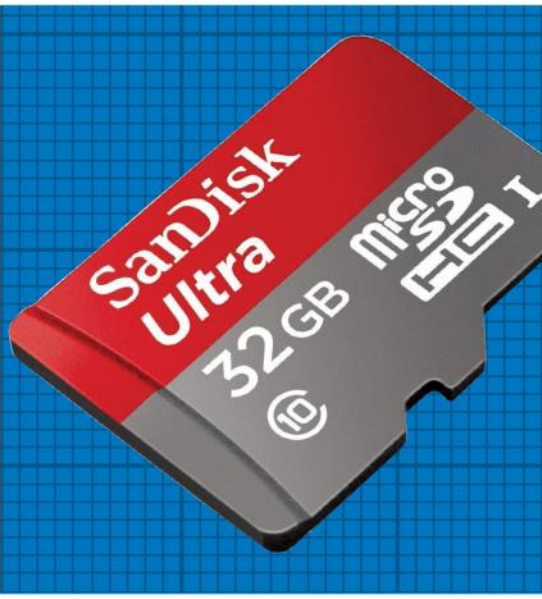
SD CARDS

The main storage for a Raspberry Pi is an SD card, typically a micro SD card, though the original Model B and Model A Pis used a full-sized SD card. On the whole, you can just run out and buy any micro SD card and it should work, but it certainly makes sense to ensure that the card you buy will not only work, but will also be large enough and as fast as possible.

Not all SD cards are made the same. There are various speed categories, such as Class 6 (which denotes a minimum 6MB/s write speed) and Class 10 (denoting 10MB/s write speeds), along with the newer and faster UHS-1/2 (Ultra High Speed). One thing to keep in mind here is that the Pi's SD controller maxes out at 25MB/s. It's also not capable of utilising the technology of the latest UHS high-speed SD cards.

With all that in mind, the key thing to look for in an SD card is to get one with fast read/write access, rather than pure throughout. It's also worth keeping an eye on the list of compatible tested SD cards at **http://elinux.org/RPi_SD_cards**.

You should also aim to get a card at least 8GB in capacity. The latest build of Raspbian – the main Pi OS – is now larger than 4GB. That said, given the price of SD cards, it makes sense to go for a 16GB or 32GB card. You need to balance the slight increase in price against the fact that a larger card is likely to be of use far longer.



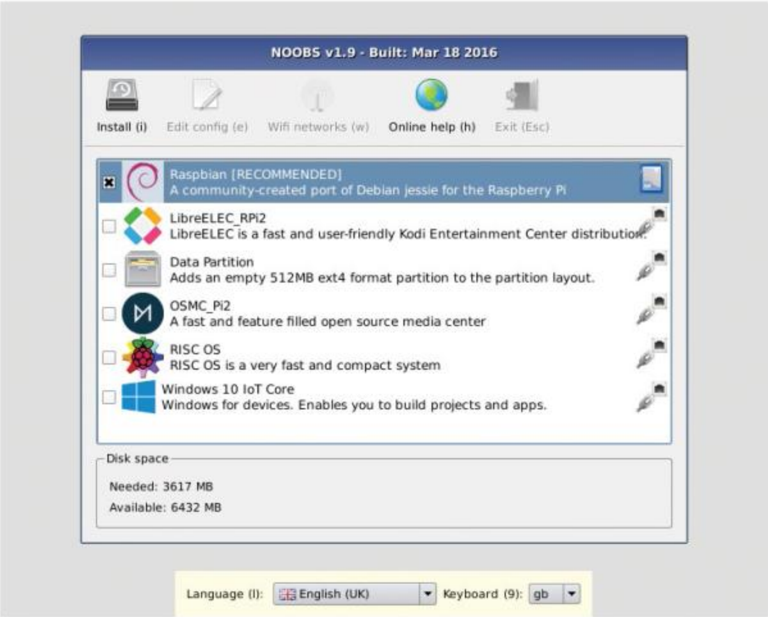
How to install Raspbian

Before you can do anything on your Raspberry Pi, you need some software for it to run. With a micro SD card in hand, we guide you through the process

Before the Raspberry Pi can do anything (apart from sitting there looking cute) it needs an operating system (OS). The most popular OS for the Pi is Raspbian, which is based on Debian Linux. A number of other operating systems are also available for the Pi. Some of these can be downloaded from the Raspberry Pi Foundation website (**https://raspberrypi.org/downloads**) and some can be installed using NOOBS (over the page). There's Ubuntu MATE (Pi 2 and 3 only), an ARM port of Arch Linux, the classic RISC OS, the OSMC media centre, not to mention the RetroPie vintage gaming platform. There's also Snappy Ubuntu Core and even Windows 10 IoT Core. If you're a beginner, Raspbian is good to start with.

Unlike traditional computers, the Pi has no internal storage, so the OS needs to be loaded from an SD card. The original Pi models (models A and B) use full-size SD cards, whereas newer models (B+, 2, 3 and Zero) use the smaller (and much easier to lose!) microSD variety. SD cards pre-loaded with an OS can be purchased from pretty much wherever the Pi is available, and are commonly included in bundles, but it is also straightforward (and cheap) to make your own. This can be done using freely available tools on Linux, MacOS or Windows.

If you already have a pre-loaded SD card, you can skip this section entirely. If not, the most straightforward way to get going with the Pi is to download the latest Raspbian image and write it directly to an SD card, which needs to be at least 4GB in capacity. If all this seems too technical, you may wish to look at NOOBS (over the page). Note that on MacOS computers, it's also possible to write the Raspbian image using the **dd** tool from the command line. If you're comfortable working this way, the procedure is very similar to the Linux instructions on page 25. Alternatively, check out the instructions on the Raspberry Pi Foundation website at **www.raspberrypi.org/documentation/installation/installing-images/mac.md**.



NOOBS v1.9 - Built: Mar 18 2016

☒ Raspbian [RECOMMENDED]
A community-created port of Debian jessie for the Raspberry Pi

☐ LibreELEC_RPi2
LibreELEC is a fast and user-friendly Kodi Entertainment Center distribution.

☐ Data Partition
Adds an empty 512MB ext4 format partition to the partition layout.

☐ OSMC_Pi2
A fast and feature filled open source media center

☐ RISC OS
RISC OS is a very fast and compact system

☐ Windows 10 IoT Core
Windows for devices. Enables you to build projects and apps.

Disk space
Needed: 3617 MB
Available: 6432 MB

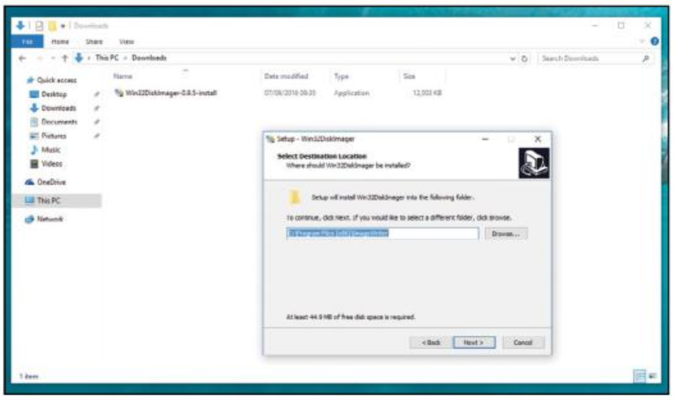
Language (l): English (UK) Keyboard (9): gb

› Raspbian should be your first choice, but the other OSes can be fetched and installed with minimum fuss.

Enjoy Raspbian

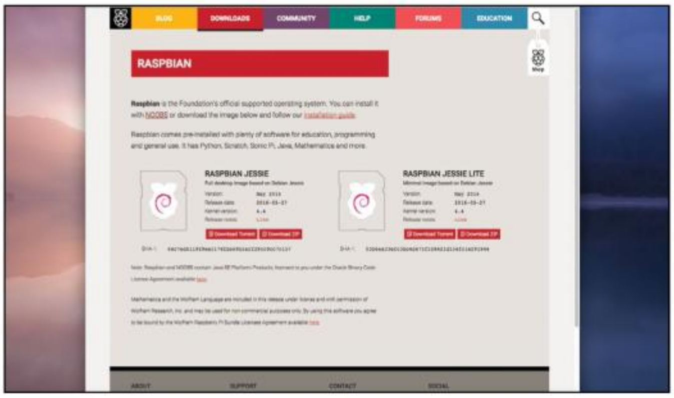
Once your SD card is ready (whichever road you choose), you're all ready to boot up your Raspberry Pi. Remove the SD card from your computer, plug it into your Pi (along with all the other gubbins – display, power, keyboard and mouse), and it should boot up to the raspi-config program. From here, you should expand the filesystem (if your card is greater than 4GB), change the default password and enable Boot to Desktop. Select Finish and you'll be able to reboot to the Raspbian desktop. There's all manner of interesting things you can play with right away, including Minecraft, Wolfram Alpha and Sonic Pi. If you have a wired network connection, then that will work out of the box. Some wireless ones will as well, but for many, this will be their first encounter with the recalcitrance of Linux. Just be patient, and remember, Google search is your friend…

INSTALL RASPBIAN WITH WINDOWS OR MACOS



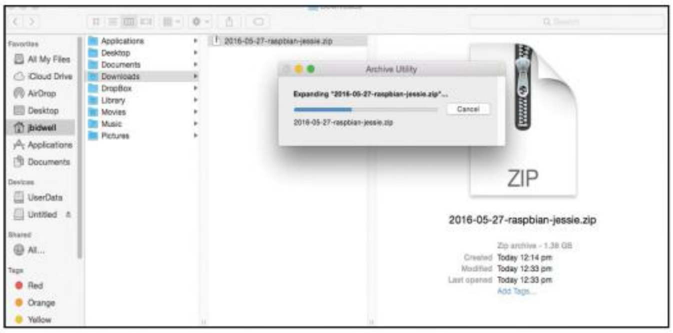
1 SD card tools

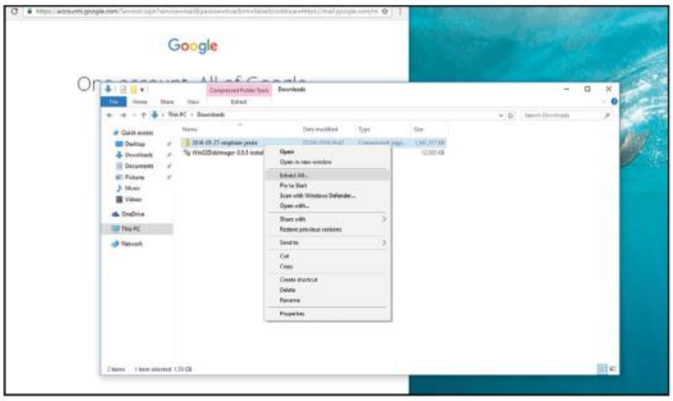
We first have to download some additional image-writing software. For Macs, the simplest tool is Rpi-sd Card Builder (**http://alltheware.wordpress.com/2012/12/11/easiest-way-sd-card-setup**), but you may also want to look at Pi Filler (**http://ivanx.com/raspberrypi**) or ApplePi-Baker (**www.tweaking4all.com/hardware/raspberry-pi/macosx-apple-pi-baker**). On Windows setups, we recommend using Win32 DiskImager, which can be downloaded from **https://sourceforge.net/projects/win32diskimager**. Installation of all of these programs is straightforward.



2 Download Raspbian image

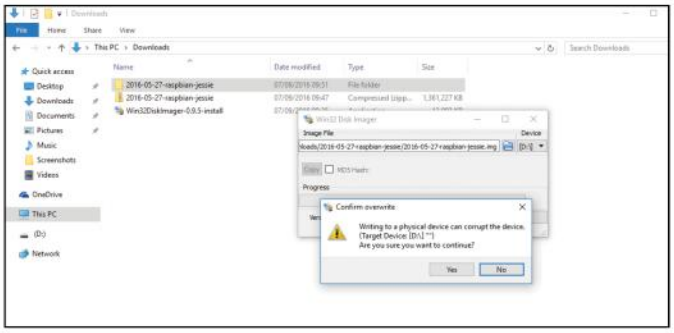
You'll find the Raspbian image at **www.raspberrypi.org/downloads/raspbian**. The current edition is called Jessie, the same as the Debian release it's based on. Be sure to choose the full edition, rather than the Lite one, at this stage – unless, of course, you're sure that you're happy at the command line, with no desktop environment. You may wish to save the Foundation some bandwidth and use the BitTorrent link. This requires you to have a BitTorrent client installed, such as Transmission or uTorrent. Whatever your download method, the full Raspbian image is a 1.3GB download, so will take a few minutes.

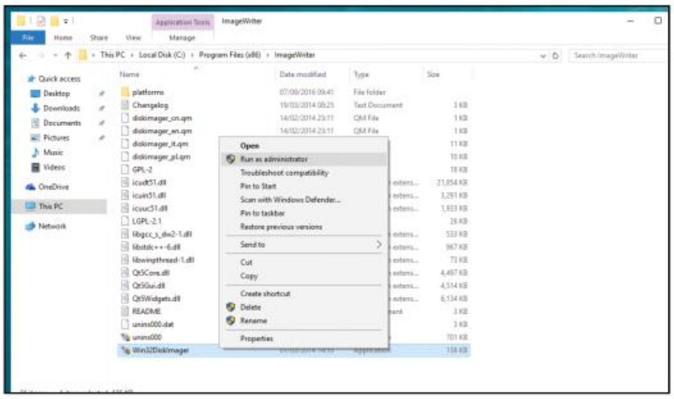




3 Unzip file

You'll end up with a file in your downloads folder, called something such as **2016-05-27-raspbian-jessie.zip**. Make sure that you're not running short of hard disk space, because this file will be close to 4GB in size when it's uncompressed. Double-click the ZIP file (if you're using a Mac), or right-click it and select Extract All (on Windows). It should extract to a file called something along the lines of **2016-05-27-raspbian-jessie.img** in the same folder as the downloaded ZIP file. This is the image file that we need write to the SD card in the next step.





4 Write the image

Win32 Disk Imager needs special privileges to write to the SD card. So, instead of double-clicking its executable, right-click it and select Run as Administrator. Now choose the image file you extracted in the previous step. In the Device section, choose the drive letter of the SD card. Click Write, double-check you have the right device, and writing commences. Wait for it to complete and, hey presto, all done. Mac tools work similarly (you may be asked for a root password), but be sure to choose the correct SD card device, rather than your hard disk. Whatever destination you choose is wiped, so you don't want to get this wrong.

Install NOOBS with any OS

Possibly the easiest way to get a new OS up and running on your Pi can be achieved using any operating system – as long as you know how!

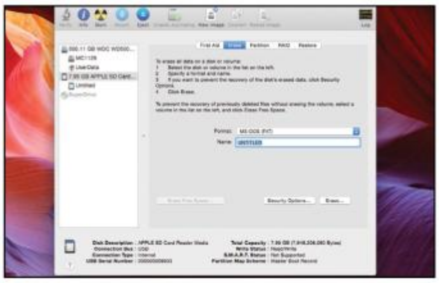
NOOBS (New Out Of the Box Software) makes it easy to install Raspbian (or other OSes) on your Raspberry Pi. You can buy an SD card with NOOBS on it from many vendors, or we'll show you how to do it yourself below.

With a correctly-prepared NOOBS SD card, the Pi can boot to a menu offering you a choice of OSes. Pick one and NOOBS installs the OS on the SD card, so it will load the next time the Pi is rebooted. Or you can hold down Shift as the Pi boots to use the NOOBS installer once again.

Preparing a NOOBS SD card with Raspbian on it requires at least a 4GB SD card – other OSes (in particular, Windows 10 IoT Core) will require 8GB. The simple process of installing NOOBS to an SD card is outlined below. Note that if you've bought an SD card with NOOBS already on it, you don't need this section – just plug in and turn on.

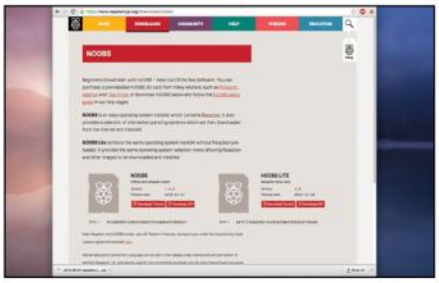
Once your card is ready, reboot the Pi with power, mouse, keyboard and monitor all connected. You should see a list of OSes (tailored for your Pi model). The Raspbian entry should have a picture of an SD card on the right, indicating that it can be installed even if the Pi is not connected to the internet. Tick the box to its left, hit Install and, before long, you'll be able to reboot straight into Raspbian.

INSTALL RASPBIAN WITH WINDOWS OR MACOS



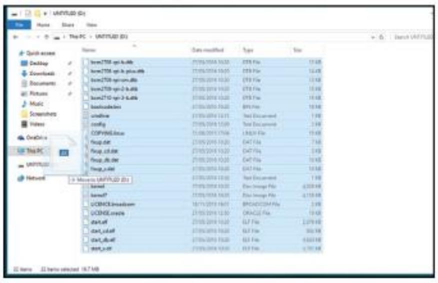
1 Format the SD card

Before you can do anything, your SD card must be correctly formatted. Freshly purchased blank cards tend to already be formatted as FAT32, but if you're recycling an old one, you need to do this yourself. First make sure there isn't anything important on it. Formatting can be done directly from Windows Explorer or from Disk Utility in MacOS. In Explorer, right-click the SD card and choose Format; in Disk Utility, go to the Erase tab. In both cases, choose FAT32, not the NTFS or HFS+ filesystems. Linux users can format SD cards using Gparted, Gnome Disks or the command line with `mkdosfs -F32 /dev/sdX1` , for example. Note that NOOBS requires that only a single FAT32 partition exists on the card, so if there are others, these should be removed with appropriate partitioning tools.



2 Download NOOBS

NOOBS is available in two forms: the regular edition (weighing in at 1GB, zipped) and NOOBS Lite (a mere 27MB). The Lite edition does not include the Raspbian image – it has to be downloaded from within the NOOBS installer. The full-fat edition allows Raspbian to be installed without an internet connection, but other OSes have to be downloaded. Again, this is easily achieved from the comfort of the NOOBS installer. Both versions can be downloaded from **www.raspberrypi.org/downloads/noobs**, either directly or via BitTorrent. The latter is a little more considerate of bandwidth but requires a BitTorrent client. Many Linux distros include the Transmission BitTorrent client (also available for MacOS), and there are plenty to choose from on Windows (uTorrent, Deluge, Bitcomet…).



3 Unzip and copy

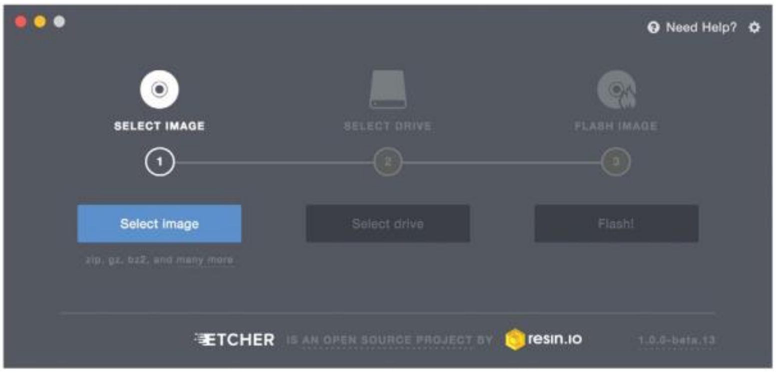
You should have a file named something like **NOOBS-V-1-9-2.ZIP** in your downloads folder. This must be decompressed and the contents put on the SD card. On Windows and Mac, you should be able to extract the files by just double-clicking the ZIP file. The files extract to a folder named something like **NOOBS-v1-9-2**. Now they must be copied to the SD card. Be sure to transfer only the files inside the NOOBS folder, rather than the folder itself, otherwise it won't work. Open the **NOOBS-v1-9-2** folder and select all the files (click the first one, hold Shift, then click the last one). Drag them on to the SD card icon and wait for them to finish copying (the full version can take a while). Right-click and safely eject the SD card. You should now be able to use this card to boot your Pi, as we'll explain later.

Install Raspbian with Linux

If you're running any flavour of Linux on your computer, then you have an even easier life when it comes to getting Raspbian on to your Pi

Linux users can get Raspbian set up from the comfort of their operating system, too. Much the same as on Windows and Mac, the first steps are to download the Raspbian image (either from a web browser or using Wget) and unzip it. There are a few different options for writing the image. You can use a graphical tool such as Etcher (see **www.etcher.io**) or Ubuntu's Startup Disc Creator. These work very much like the image-writing tools for MacOS and Windows. As with most tasks on Linux, the usual approach is to use the command line.

Insert the target SD card into the machine. We'll use the **dd** tool, which is part of all standard Linux installations, to transfer the image. You definitely don't want to accidentally wipe your hard drive this way (and it's possible to do just that if you get this wrong), so first use the `lsblk` command to see which device node it is attached to. You should be able to identify the SD card by its capacity. It could be **/dev/sdb** or it could be **/dev/sdc**. Newer PCs that have a native MMC controller will give it a slightly different name, such as **/dev/mmcblk0p1**. Note that it is the device, not partitions on that device (such as **/dev/sdb1** or **/dev/mmcblk0p1n2**), that we are interested in. Any existing partition information will be lost when we transfer the image to the device. Writing the image to a partition won't work at all well, so don't do that.



❮ The *Etcher* image writing program is probably the slickest way to make SD cards under Linux, and saves you from getting your hands dirty at the command line.

We'll refer to the SD card as /dev/sdX (replace the X with the actual drive letter) and assume you've unzipped the Raspbian IMG file to the Downloads directory. In a terminal type:

```
$ sudo dd if=~/Downloads/26-05-16-raspbian.img of=/dev/sdX bs=1M
```

This process can take a long time, particularly on older or cheaper SD cards. Sometimes, even when the command completes (when you are able to enter another command), the job is not done. If your SD card or adaptor has an activity light, then you can see if anything's still being transferred, but if not, try the `sync` command. If it doesn't complete right away, data is still in-flight. Once everything is ready, you can remove the SD card, boot your Pi and tweak the raspi-config program as per the Windows and MacOS instructions.

Linux luck

On Linux, there is a slight advantage in that the SD card can be browsed natively. It's only a slight advantage because things can be sent over the network to side-step filesystem incompatibilities amongst OSes. Still, it can sometimes be handy (for example, if the Pi is shut down) to just be able to grab the SD card, slip it in a slot, and grab photos or edit configuration files.

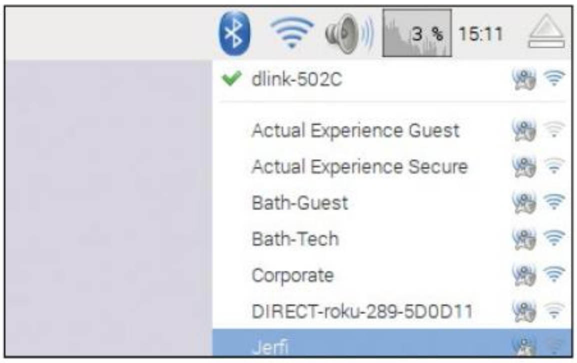
If you're not (yet) a Linux user, it's also possible to use the live disc mode of a distribution (Ubuntu, Fedora, openSUSE, Mint, for example) and create the SD card from there. There's a slight problem downloading the image to the root filesystem, however, because this is all in RAM for live discs (their modus operandi is to not touch your hard drive), and there typically isn't enough space for several gigabytes of OS image. This can be worked around, though, because the image can be downloaded to a USB stick beforehand and copied from there. On Ubuntu, USB sticks are mounted in a directory inside **/run/media/ubuntu**, so change the `if` parameter of the `dd` command accordingly. Alternatively, you can use the Startup Disk Creator on an Ubuntu live DVD if you don't feel like typing things.

Using networks on the Pi

Getting hooked up to a local network and the wider internet is easier than you might think, as long as you have all the right bits...

Everyone (and, it seems, everything) is online these days, and that means that if you want your Raspberry Pi to do anything "interesting" – even if that's just browsing Facebook – you need to get it networked. As with everything Raspberry Pi, it's made as hard as possible, to ensure you learn as much as possible along the way! No, actually, the only complication is that Raspbian doesn't hide any of the nitty gritty of networking from you, combined with the need to add wireless dongles to all Raspberry Pi models except the latest Pi 3, which comes with built-in wireless and Bluetooth.

Let's talk wireless adaptors. The first thing you need to do before buying a wireless adaptor is to check whether the one you already have or plan to get will work with your Raspberry Pi. That's where **http://elinux.org/RPi_USB_Wi-Fi_Adapters** comes in – this is a huge list of both verified adaptors and those that cause issues.



Raspbian now supports easy GUI-based wireless connections – so modern!

The good news is that a huge range works with Raspbian, thanks to its Linux kernel. It comes with built-in support for a large number of adaptors. Because of the way Raspbian is built, an adaptor that doesn't work with Raspbian might work with another Linux OS, such as Arch or OpenELEC, because of the particular version of the Linux kernel they ship with.

You can confirm whether a USB device is working correctly – other than the Raspbian wireless being enabled – by opening a terminal and typing `lsusb` , then pressing Return. This provides a list of all your USB devices, and one should be something that sounds like your USB Wi-Fi device.

A second command, `iwconfig` provides a list of working wireless adaptors connected to your Pi. This offers a slightly more direct confirmation that the USB dongle is connected and working.

An easy way to avoid any worry is to buy an official Raspberry Pi Foundation Wi-Fi dongle. Designed to work with all Raspberry Pis and Raspbian, this eliminates any worries. There's also the added point that it helps fund the Foundation, too. You can get more details from **www.raspberrypi.org/products/usb-wifi-dongle/**.

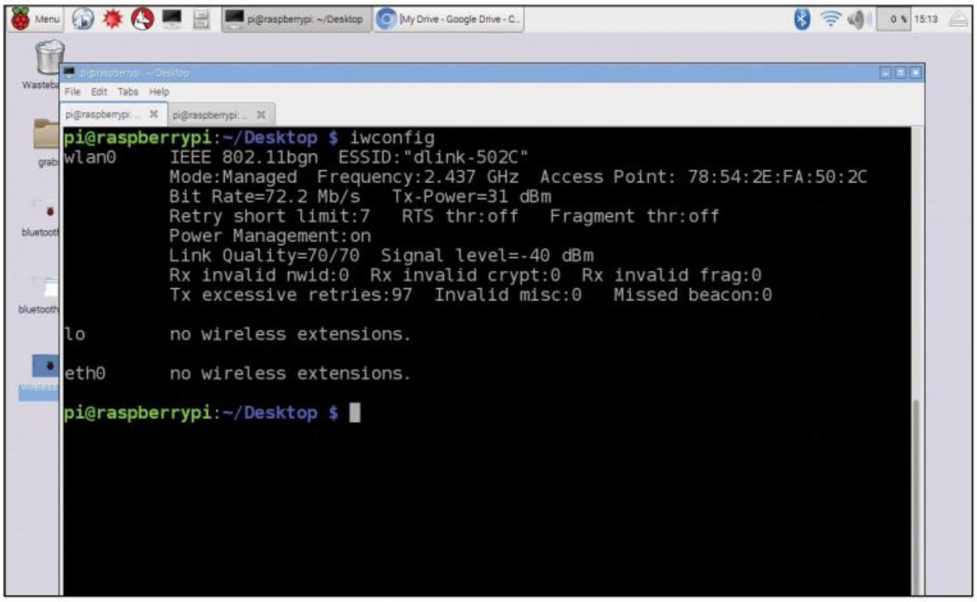
Back in the day, you had to edit text files to configure your wireless dongle, but these days, Raspbian has been fully updated to provide a GUI for connecting to any wireless network. How civilised! Wi-Fi connections can be made via the familiar network icon at the right-hand end of the menu bar. If a working Wi-Fi dongle is plugged in, left-clicking this icon brings up a list of available Wi-Fi networks, as shown in the picture on this page. If no networks are found, it shows the message "No APs found – scanning..." Wait a few seconds more without closing

TCP AND YOUR SORE IP

If you're new to PCs and networking, then the sudden talk of IP numbers and the like might be very confusing. Let's just say that local networks and the internet all use a very clever system called TCP/IP. On a pedantic level, that's two – yep, two – communications protocols: TCP, alias Transmission Control Protocol; and the IP part, alias Internet Protocol. You don't really need to know anything about it, other than this: it assigns a unique number to each and every device on the internet. This enables messages to get to and from your devices, finding their way to the right place.

Each device gets an address made up of four numbers, separated by full stops. Each number can be between 0 and 255. So, typically, an address would be something like 192.168.0.1 or 192.168.100.23. Generally, your home and business network has its own IP range – the broadband router you use to connect to the internet creates this automatically using something called a DHCP server, and uses Network Address Translation, or NAT, to route messages to and from your local network and the wider internet. See, it does get awfully complicated very quickly!

Usually, when we talk about IP addresses, it's when you want to know the address of a device or server on your network or on the internet, or you want to give a device a specific address. You won't have to know much more than this.



```
pi@raspberrypi:~/Desktop $ iwconfig
wlan0     IEEE 802.11bgn  ESSID:"dlink-502C"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 78:54:2E:FA:50:2C
          Bit Rate=72.2 Mb/s   Tx-Power=31 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=70/70  Signal level=-40 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:97  Invalid misc:0   Missed beacon:0

lo        no wireless extensions.

eth0      no wireless extensions.

pi@raspberrypi:~/Desktop $
```

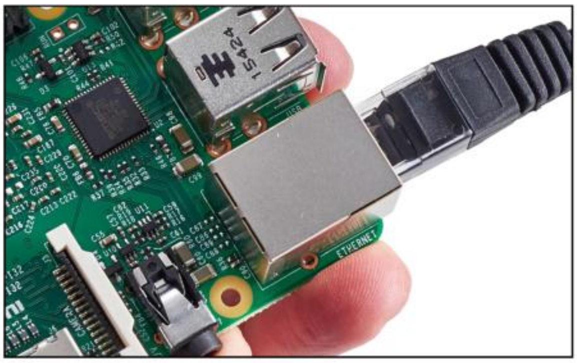
The iwconfig command helps check whether a wireless adaptor is working.

the menu, and it should find your network.

The icons on the right show whether a network is secured or not, and its signal strength. Click the network that you want to connect to; if it is secured, a dialog box appears, prompting you to enter the network key. Do so and wait a couple of seconds. The network icon flashes briefly to show that a connection is being made; once it is ready, the icon stops flashing and shows the signal strength.

Older versions of Raspbian provided a Wi-Fi Config tool on the desktop. There's no reason why you should encounter an older version at this point, but if you do, it's no more complicated than running it and then using its Scan button to locate your desired wireless network.

Wired networking is the fastest and most reliable way to connect things.



Wired networks

We should also mention wired networks while we're here. All Model B Raspberry Pis have a wired network connection called an Ethernet port. This uses an Ethernet cable to connect to a standard router. A wired connection remains the most reliable and the fastest option for transferring files and getting an internet connection. In terms of having to do anything, you don't – just plug in the cable, and as long as it's connected to a working router that already has an internet connection, Raspbian is allocated an IP address.

Most Ethernet ports provide two status LEDs (the original Pi did not; these status lights were moved to the board itself). The green confirms a working connection (any blinking shows activity), while the amber one announces a full-speed 100Mbs connection (or 10Mbs if it's off).

It's also worth noting that the Linux kernel can happily handle multiple network connections over different network adaptors. This means you can connect to a different network or the same one via both a wired connection and wireless at the same time – the kernel's TCP library happily balances transfers over both the adaptors. It also means you can remotely connect to the Pi via either assigned IP address.

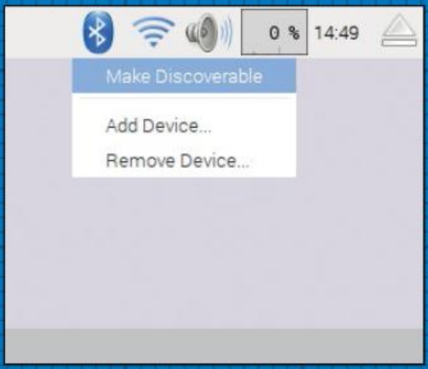
As with USB Wi-Fi dongles, it's also possible to add wired Ethernet adaptors via a USB port – a full list of verified and problem adaptors can be found at **http://elinux.org/RPi_USB_Ethernet_adapters**. While this doesn't make total sense for a Model B board, it may be exactly what you need when it comes to the Model A or the Pi Zero.

The subject of networking can get highly involved, so we're going to leave things with a standard IP connection.

BLUETOOTH DEVICES

With the addition of the embedded dual wireless/Bluetooth abilities in the Pi 3, Raspbian was updated to support Bluetooth out of the box. Prior to this, you had to "hack" in the Bluetooth support, with the command `sudo apt-get install --no-install- recommends bluetooth` and, once it was installed, use `sudo service bluetooth status` to enable it.

While this was a simple case of installing a manager tool, it's far neater having it around by default. If you use a smartphone you're likely to have used Bluetooth, with all the Bluetooth speakers, keyboards and headsets available these days, but if not, the good news it that's it's far simpler than connecting to a wireless network. This goes for all Bluetooth devices, but you first need to make them "discoverable" – this enables other Bluetooth devices to then attempt to connect with them, at which point you have to allow those connections or not, as the case may be. Left-click the Bluetooth icon in the top-right of the Raspbian desktop, select Make Discoverable, then choose Add Device to pull up a list of nearby Bluetooth devices. Select the correct one and click Pair. With keyboards and devices, you often have to enter a code on the device to finalise the connection. You'll want to ensure that you choose Stop Discoverable afterwards.



The Raspberry Pi should pair with most Bluetooth devices without an issue.

Master the new Pixel Desktop

We explore the new Pixel desktop bundled with the latest version of Raspbian, and show you how to update your existing installation

Christmas has come early to the Pi this year with Pixel (Pi Improved Xwindows Environment, Lightweight). If you think of the former desktop environment LXDE as a generic store-bought plastic tree, by contrast Pixel is a towering Norwegian Pine laden with tinsel and toys.

Pixel's perfection is evident from the moment you switch on any Raspberry Pi with the latest install of Raspbian as the scrolling line of (largely obscure) text has now been replaced with an elegant splash screen, displaying only a few helpful messages and the version number. (The splash screen can however be reversed or even switched off altogether – see *Climb out of the Splash Screen* on p33 for more information).

The real treat comes when the desktop loads. Thanks to Pi Foundation Developer and skilled photographer Greg Annandale, there are now over a dozen epic desktop backgrounds bundled with Raspbian. Our current favourite is "Mountain", but they're all stunning.

The Pixel cheer continues upon opening a window, as you will notice that the wide frames around the borders are now gone, giving it a much more contemporary look and feel reminiscent of an actual desktop computer. Mercifully it is still fairly easy to resize the window (see "NitPixels," p31).

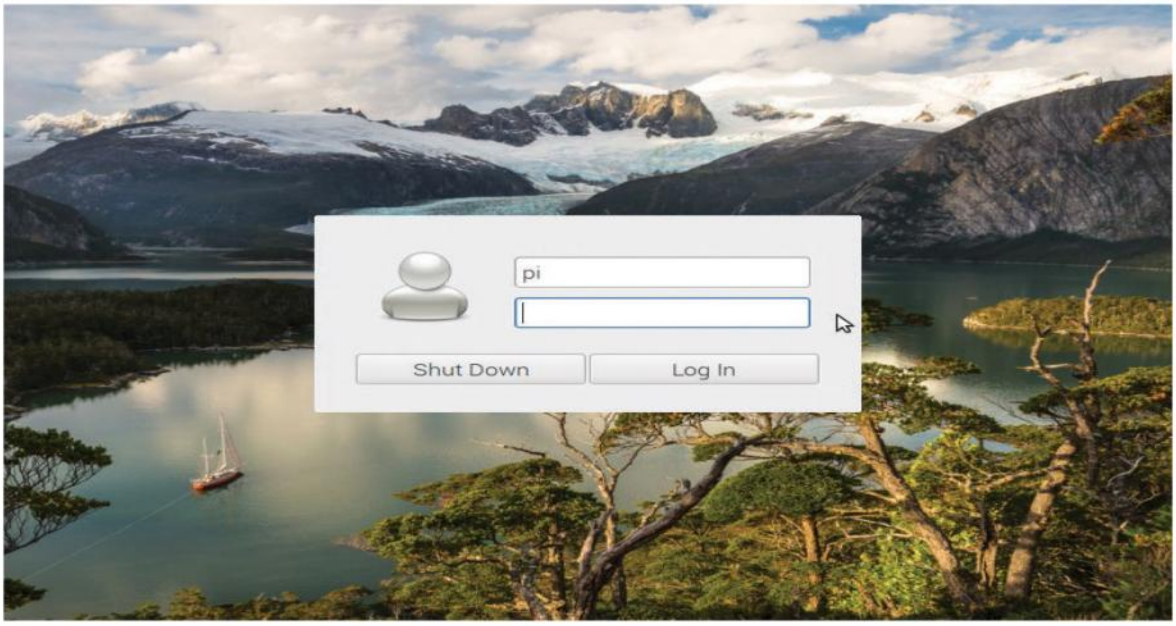
If the windows alone are not enough of a visual feast, two of the Pi Foundation's web developers have spent a great deal of time on making funky yet functional icons for the Pixel desktop. You can open the applications menu to see most of these but they excited us so greatly we actually headed over to the Add/Remove Software section in order to see the full set for various categories.

Some of the changes to the desktop environment are much

SWITCH ON, AND THE SCROLLING TEXT HAS BEEN REPLACED WITH AN ELEGANT SPLASH SCREEN

Quick tip

All system icons are stored in **/usr/share/icons/PiX**. Feel free to replace them with your own icon set, but keep the filenames and sizes the same.



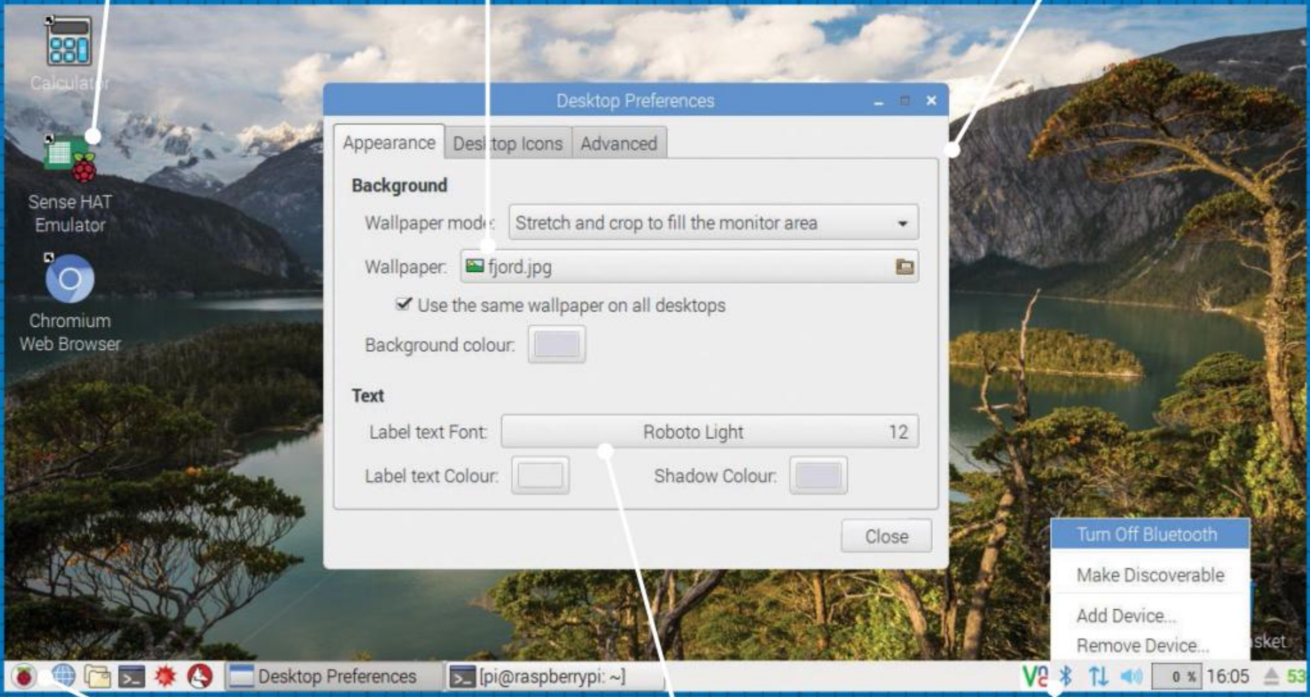
› **By default Raspbian still doesn't require a password. You can, however, enable the login screen, which will appear on your chosen desktop background.**

HIGHLIGHTS OF THE PIXEL DESKTOP

Applications New apps include the Chromium Browser, the SenseHAT Emulator and RealVNC server. See the main text for more information.

Wallpaper The desktop now has a choice of 16 separate stunning wallpapers. Visit your appearance settings as shown to change these.

Windows The clunky frame around windows is gone. You can still resize windows by hovering your mouse pointer just outside a window edge.



Background

Wallpaper mode: Stretch and crop to fill the monitor area

Wallpaper: fjord.jpg

☑ Use the same wallpaper on all desktops

Background colour:

Text

Label text Font: Roboto Light 12

Label text Colour:

Shadow Colour:

Close

Icons Icons on the taskbar, menu and file manager have been redrawn in colourful and painstaking detail.

Integrated fonts The Infinality font rendering package is now included to optimise how fonts are mapped to Pixels on the screen.

Interfaces Wireless and Bluetooth icons now allow for power switching, in that one or both can be disabled from within the desktop.

more subtle. The Roboto font, which is still the default font for Raspbian, has been augmented with a rendering package to map pixels to the screen. The effect is admittedly not very noticeable in many places but makes for a much more professional feel in others.

If you are unswayed by Pixel's glittery showing then there are also several titanic applications bundled with the latest version of Raspbian. Chief among these is integrated support for VNC out of the box (see *As Easy as RealVNC*, p31, for details).

Raspbian+Pixel also now includes an "initial release" of Chromium for the Pi. If you're not familiar with it, Chromium is the open source project behind the Google Chrome web browser. For more information, visit **www.chromium.org**.

Playing with Pixel

Hardware geeks may also be delighted at the inclusion of a SenseHAT emulator which allows for easy testing of interfacing projects without having to buy an actual SenseHAT board or reconfigure one each time.

The Pixel Desktop will now be bundled with all future installs of Raspbian and we encourage you to try it out. The simplest way if you use Raspbian with NOOBS, the easy operating system installer, is to restart your Pi while holding Shift and select Raspbian again. Be warned, however, that this will erase your existing installation.

Alternatively, you can choose to download the latest Raspbian image from the Raspberry Pi website (**www.raspberrypi.org/downloads/raspbian/**) and then follow the installation »

Quick tip

If you want to port any of the sample code from the emulator to a real SenseHAT, simply change **sense_emu** to **sense_hat** at the top of your program.

Quick tip

Consider installing the plugin Disconnect from the Chrome Web Store. This blocks websites from tracking you and can significantly increase your browsing speed.

guide to move it onto a suitable MicroSD card from which you can boot your Pi (**http://bit.ly/Pi-install-guide**).

Thankfully it's also possible to upgrade your existing Raspbian Jessie to the latest version including Pixel (Raspbian+Pixel). See our step-by-step guide on p32 for more information.

Chromium comes along

Not only does Chromium provide a much more modern look and feel to everyday web browsing on the Pi, it's also optimised to stream HD video with your Pi's hardware. This is achieved in part by the h264ify browser extension, which as the name suggests forces YouTube to stream hardware accelerated H.264 videos.

The Ublock Origin extension is also installed to prevent the browser being slowed down by annoying ads. Ublock Origin is based on the more well-known AdBlock Plus but unlike its big brother filters all advertising content as well as trackers rather than let a few supposedly harmless adverts through.

Once it's installed the web browser button will launch Chromium by default. However, you can launch the former browser Epiphany at any time by running the command `epiphany-browser`

You may want to do this if, for example, you want to export your old bookmarks into Chromium (see "NitPixels" on the opposite page for more).

The official Raspberry Pi website cautions that although Chromium runs well on the Pi 2 and 3, it might struggle on older models and the Pi Zero. If this applies to you, you might consider using Epiphany or investing in a faster Pi.

A word on interfaces

By popular demand, the latest iteration of Raspbian gives you much more control over your interfaces. It's finally possible to disable either Bluetooth or Wi-Fi altogether from inside the Pixel desktop environment. Simply click the Bluetooth/Wi-Fi icon and choose to disable either or both. This is particularly useful if you're running your Pi from a battery pack (because both of these are power-hungry and are best switched off if not actively in use) or you're storing sensitive data and don't want to be visible on networks.

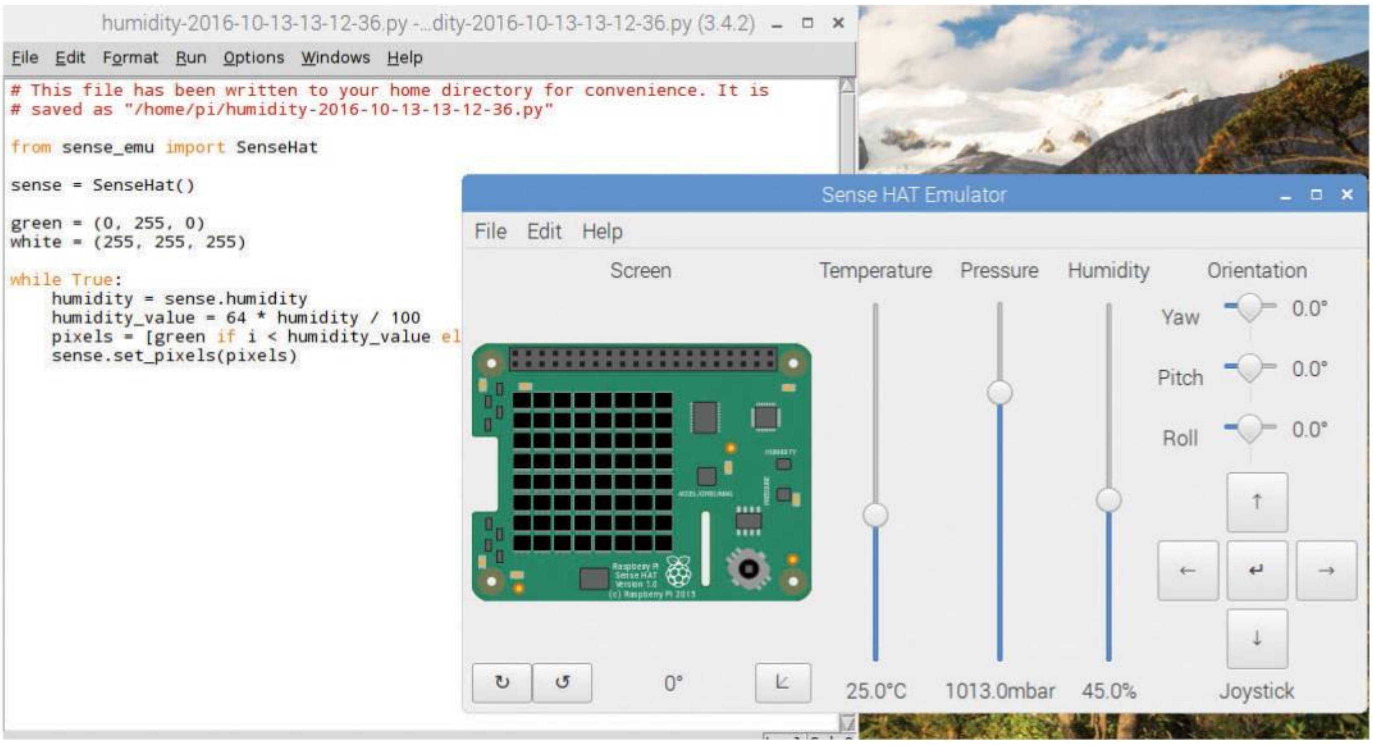
If you do choose to leave the Wi-Fi enabled, you can now also disconnect from your currently connected wireless network with two clicks of your mouse. Simply open the network menu and click on your current network name to disconnect.

Sensational SenseHAT

For the benefit of those people who haven't (yet) visited the International Space Station, the SenseHAT is an add-on board for the Raspberry Pi. In November last year it was used as part of the Astro-Pi project (see **https://astro-pi.org** for more information) and is now available on sale from sites like the Pi Hut for around £30.

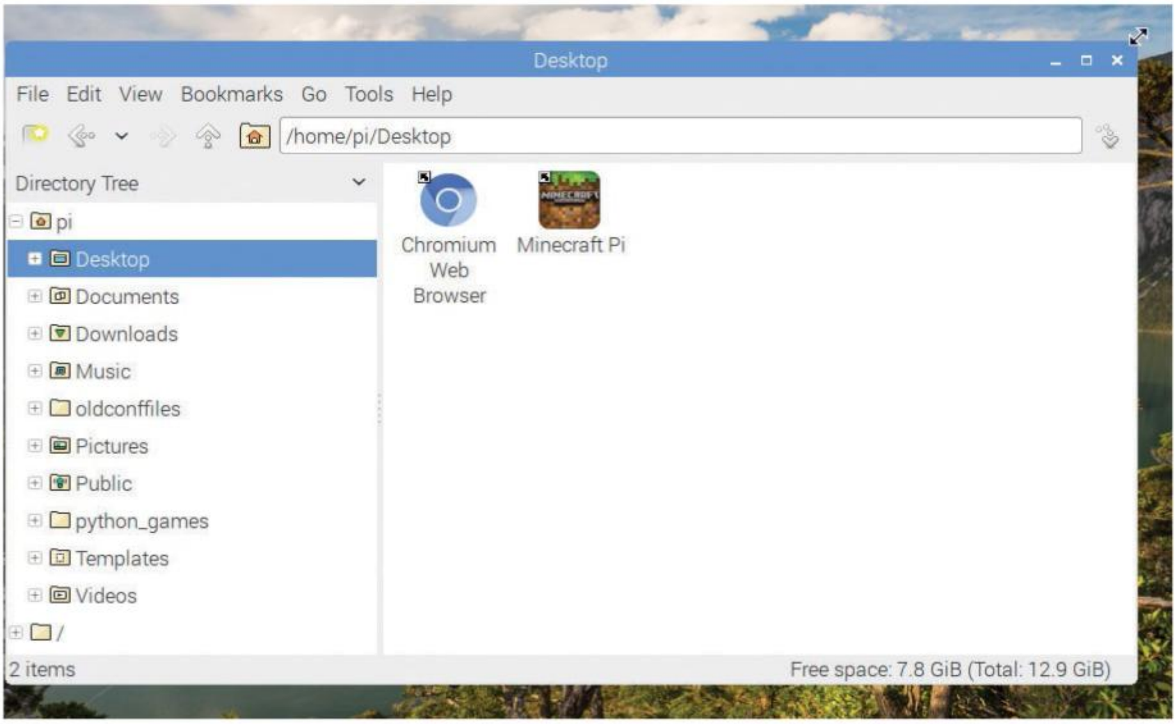
The SenseHAT is suitable for much more than space satellites because it's bristling with useful sensors including a gyroscope, accelerometer, magnetometer,

THE EMULATOR ENABLES YOU TO TEST ALL YOUR PROJECTS BEFORE BUYING A SENSEHAT BOARD



```
humidity-2016-10-13-13-12-36.py -...dity-2016-10-13-13-12-36.py (3.4.2)
# This file has been written to your home directory for convenience. It is
# saved as "/home/pi/humidity-2016-10-13-13-12-36.py"

from sense_emu import SenseHat

sense = SenseHat()

green = (0, 255, 0)
white = (255, 255, 255)

while True:
    humidity = sense.humidity
    humidity_value = 64 * humidity / 100
    pixels = [green if i < humidity_value el
    sense.set_pixels(pixels)
```

The SenseHAT Emulator uses click and slide buttons to emulate movement of the joystick and changes in the environment. This example program changes the LED colour based on humidity.



/home/pi/Desktop

2 items

Free space: 7.8 GiB (Total: 12.9 GiB)

Move the mouse pointer either to the edge of a window or just outside it and you can resize the window.

thermometer and barometer. It also comes complete with an 8x8 LED interface and a five button joystick.

The latest version of Raspbian+Pixel now comes with a SenseHAT Emulator, allowing budding coders and interfacers to test all of their projects before taking the plunge and buying an actual SenseHAT board.

Although the Raspberry Pi Foundation previously announced a web based emulator for the SenseHAT, this one has been developed by the awesome Dave Jones to run natively on the Pi. Environmental variables like temperature and pressure can be altered using slider switches.

For those who are interested in getting started, the emulator contains a number of example projects. Launch the emulator, then click the File menu at the top left, then hover over Open Example. You will see there are projects suitable for beginners, as well as intermediate and advanced examples.

NitPixels

The upgrade process is reasonably pain free but if you choose to install Raspbian+Pixel from scratch, the ZIP file will decompress from 1.8GB to just over 4GB in size. This shouldn't pose a problem for more recent versions of Linux but if you are using a different OS, older

»

AS EASY AS REALVNC

RealVNC have brought both their viewer and server software to the Pi, which makes life much easier for those who want to connect remotely to the rich GUI.

Once RealVNC is installed, go to Menu > Preferences > Raspberry Pi Configuration > Interfaces and click "Enabled" under VNC. The VNC icon will appear at the top right of the desktop. Click this to open the VNC window complete with a unique IP address for VNC clients – for example **http://192.168.1.10**.

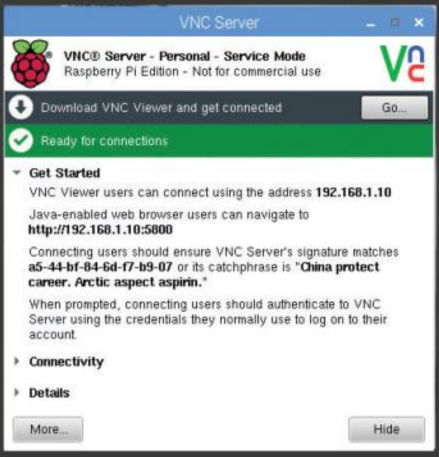
Alternatively if you're using a Java enabled web browser, you can choose to use the relevant link to open a separate Java window inside which you can access the Pi's desktop from another computer without installing additional programs.

If you choose to do this, first visit **https://java.com/en/download/installed8.jsp** to make sure that your browser does indeed support Java. If not, you'll be given the opportunity to install it immediately.

Linux users may prefer to use the IcedTea-Web plugin for your browser. The software is open source and is capable of running Java applets. Visit **http://icedtea.classpath.org/wiki/IcedTea-Web** for information on how to download and install the latest version.

Whichever method you use, you'll be asked to click "Run" to confirm that you wish to use the RealVNC software. The desktop will open in a new window. Be careful not to close this window (we recommend moving it to a separate desktop if your system supports this) because if you do, you'll lose access.

If you plan to use VNC regularly, you might wish to consider bookmarking the Java link so that you can open your desktop at any time.


VNC Server
VNC® Server - Personal - Service Mode
Raspberry Pi Edition - Not for commercial use
Download VNC Viewer and get connected
Go...
Get Started
Connectivity
Details
More...
Hide

Ready for connections

VNC Viewer users can connect using the address **192.168.1.10**

Java-enabled web browser users can navigate to **http://192.168.1.10:5800**

Connecting users should ensure VNC Server's signature matches **a5-44-bf-84-6d-f7-b9-07** or its catchphrase is "**China protect career. Arctic aspect aspirin.**"

When prompted, connecting users should authenticate to VNC Server using the credentials they normally use to log on to their account.

Note down the IP address for your VNC client/web browser. Use the security catchphrase to stop your connection from being intercepted by "Man in the Middle" attacks.

archiving utilities may struggle with this. To avoid this issue, the Raspberry Pi Foundation recommends using 7Zip for Windows and The Unarchiver for Mac to extract the files.

For the sake of compatibility Raspbian may overwrite some of your configuration files. If you find that this interferes with anything and need to revert to the previous state, your old configuration files can be found in **/home/pi/oldconffiles**.

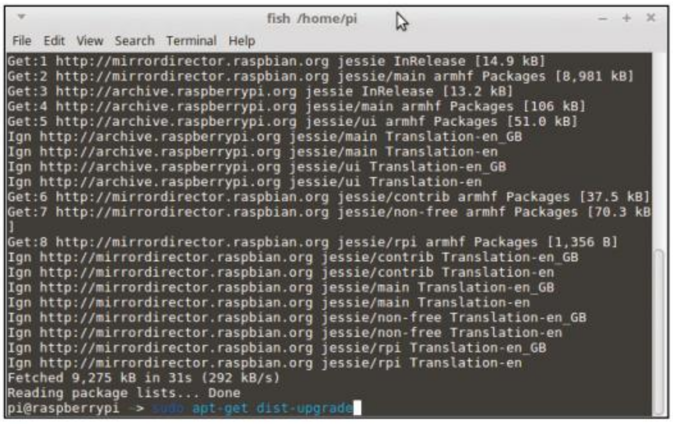
As shiny as Chromium is, before riding it eternal (we can never resist references to *Mad Max*) you may wish to import your old bookmarks from the previous Epiphany browser. You can launch Epiphany at any time by opening Terminal on the Pi and running the command `epiphany-browser` Click the settings gear icon on the top right and then choose Edit Bookmarks. A new window will open, giving you the chance to export your bookmarks in HTML format.

The first time you run the Chromium browser you'll see a blue link offering to Import Bookmarks. Click this and locate the file to bring your favourites across to the new browser. Bear in mind that this version of Chromium is experimental and therefore not all websites may perform as you'd like.

Getting used to it

Resizing windows now that the oversized frames are gone can take a little getting used to. The resize area actually exists just outside the window as well as on the very edge, so you don't

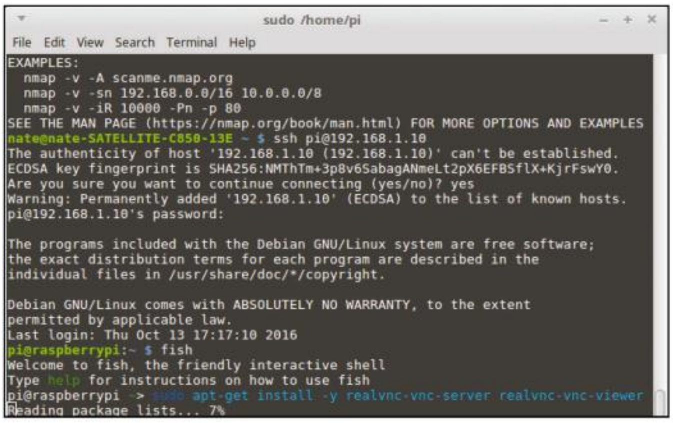
UPDATING AN EXISTING INSTALLATION



```
Get:1 http://mirrordirector.raspbian.org jessie InRelease [14.9 kB]
Get:2 http://mirrordirector.raspbian.org jessie/main armhf Packages [8,981 kB]
Get:3 http://archive.raspberrypi.org jessie InRelease [13.2 kB]
Get:4 http://archive.raspberrypi.org jessie/main armhf Packages [106 kB]
Get:5 http://archive.raspberrypi.org jessie/ui armhf Packages [51.0 kB]
Ign http://archive.raspberrypi.org jessie/main Translation-en_GB
Ign http://archive.raspberrypi.org jessie/main Translation-en
Ign http://archive.raspberrypi.org jessie/ui Translation-en_GB
Ign http://archive.raspberrypi.org jessie/ui Translation-en
Get:6 http://mirrordirector.raspbian.org jessie/contrib armhf Packages [37.5 kB]
Get:7 http://mirrordirector.raspbian.org jessie/non-free armhf Packages [70.3 kB
]
Get:8 http://mirrordirector.raspbian.org jessie/rpi armhf Packages [1,356 B]
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en
Ign http://mirrordirector.raspbian.org jessie/main Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/main Translation-en
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en
Fetched 9,275 kB in 31s (292 kB/s)
Reading package lists... Done
pi@raspberrypi ~> sudo apt-get dist-upgrade
```

1 Run Update and Upgrade

Open Terminal on your Pi or connect via SSH and run the command `sudo apt-get update` then `sudo apt-get dist-upgrade` Feel free to make yourself a hot beverage while you wait for the upgrade process to finish. Keep an eye on the process, however, as you may see an alert about the plymouth I/O multiplexing framework. Press **q** to close this and proceed.



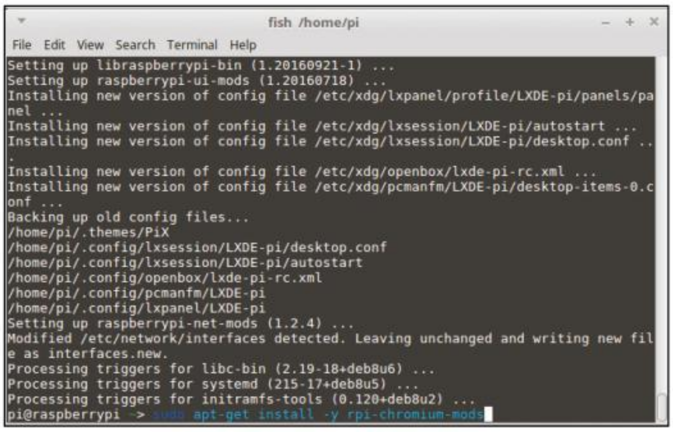
```
EXAMPLES:
  nmap -v -A scanme.nmap.org
  nmap -v -sn 192.168.0.0/16 10.0.0.0/8
  nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
nate@nate-SATELLITE-C850-13E ~ $ ssh pi@192.168.1.10
The authenticity of host '192.168.1.10 (192.168.1.10)' can't be established.
ECDSA key fingerprint is SHA256:NMThTm+3p8v6SabagANmeLt2pX6EFBSflX+KjrFswY0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.10' (ECDSA) to the list of known hosts.
pi@192.168.1.10's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct 13 17:17:10 2016
pi@raspberrypi:~ $ fish
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
pi@raspberrypi ~> sudo apt-get install -y realvnc-vnc-server realvnc-vnc-viewer
Reading package lists... 7%
```

2 Install RealVNC viewer and server

Once the upgrade is complete, install the RealVNC software with the following command:

`sudo apt-get install -y realvnc-vnc-server realvnc-vnc-viewer`

Next time the Pi restarts you'll see the VNC icon at the top right. See *As Easy as RealVNC* on the previous page for more information.



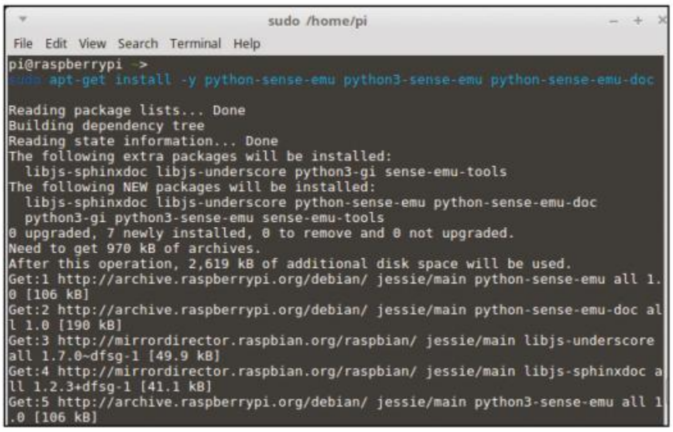
```
Setting up libraspberrypi-bin (1.20160921-1) ...
Setting up raspberrypi-ui-mods (1.20160718) ...
Installing new version of config file /etc/xdg/lxpanel/profile/LXDE-pi/panels/pa
nel ...
Installing new version of config file /etc/xdg/lxsession/LXDE-pi/autostart ...
Installing new version of config file /etc/xdg/lxsession/LXDE-pi/desktop.conf ..
.
Installing new version of config file /etc/xdg/openbox/lxde-pi-rc.xml ...
Installing new version of config file /etc/xdg/pcmanfm/LXDE-pi/desktop-items-0.c
onf ...
Backing up old config files...
/home/pi/.themes/PiX
/home/pi/.config/lxsession/LXDE-pi/desktop.conf
/home/pi/.config/lxsession/LXDE-pi/autostart
/home/pi/.config/openbox/lxde-pi-rc.xml
/home/pi/.config/pcmanfm/LXDE-pi
/home/pi/.config/lxpanel/LXDE-pi
Setting up raspberrypi-net-mods (1.2.4) ...
Modified /etc/network/interfaces detected. Leaving unchanged and writing new fil
e as interfaces.new.
Processing triggers for libc-bin (2.19-18+deb8u6) ...
Processing triggers for systemd (215-17+deb8u5) ...
Processing triggers for initramfs-tools (0.120+deb8u2) ...
pi@raspberrypi ~> sudo apt-get install -y rpi-chromium-mods
```

3 Install Chromium intro version

To install the introductory version of Chromium run the command:

`sudo apt-get install -y rpi-chromium-mods`

The Chrome plugins h264ify and uBlock Origins are included in this install, so you don't need to add them manually. During the install process you may need to hit **Y** to continue with loading Adobe Flash Player onto the Pi.



```
pi@raspberrypi ->
sudo apt-get install -y python-sense-emu python3-sense-emu python-sense-emu-doc

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libjs-sphinxdoc libjs-underscore python3-gi sense-emu-tools
The following NEW packages will be installed:
  libjs-sphinxdoc libjs-underscore python-sense-emu python-sense-emu-doc
  python3-gi python3-sense-emu sense-emu-tools
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 970 kB of archives.
After this operation, 2,619 kB of additional disk space will be used.
Get:1 http://archive.raspberrypi.org/debian/ jessie/main python-sense-emu all 1.
0 [106 kB]
Get:2 http://archive.raspberrypi.org/debian/ jessie/main python-sense-emu-doc al
l 1.0 [190 kB]
Get:3 http://mirrordirector.raspbian.org/raspbian/ jessie/main libjs-underscore
all 1.7.0~dfsg-1 [49.9 kB]
Get:4 http://mirrordirector.raspbian.org/raspbian/ jessie/main libjs-sphinxdoc a
ll 1.2.3+dfsg-1 [41.1 kB]
Get:5 http://archive.raspberrypi.org/debian/ jessie/main python3-sense-emu all 1
.0 [106 kB]
```

4 Apply all changes

Whether or not you choose to install the emulator, you should also restart the Pi at this stage to apply all changes with the command `sudo reboot` . Note the understated Raspberry Pi splash screen as you do so.

need to practise extremely precise motor control. Simply move the mouse pointer to the corner of a window and wait a moment for the cursor to change to a resize icon.

The Raspberry Pi website cautions that those using xrdp to access their Pi remotely might find that it conflicts with RealVNC. If you're updating your existing installation of Raspbian, skip step two in the step-by-step guide on the opposite page to avoid installing VNC. If you've installed Raspbian+Pixel by downloading a fresh image from the internet or you worked your way through the tutorial before reading this warning, you can run the following command to remove RealVNC server and all related files:

```
sudo apt-get purge realvnc-vnc-server
```

Finally, a lot of time and effort has gone into making Raspbian into a much better approximation of a desktop computer. If you decide to start using a Pi as a home or office computer, consider enabling the login screen to make sure anyone else accessing it will need a password. Simply run the following command to open up your desktop configuration:

```
sudo nano /etc/lightdm/lightdm.conf
```

Now comment out the line "autologin-user=pi" by placing a # at the start. Press Ctrl + X,Y, then Return to save and exit and restart the Pi. If you do this, you'll notice another neat feature of Pixel is that your login screen now appears over your existing desktop background. ■

CLIMB OUT OF THE SPLASH SCREEN

Some users may prefer the old scrolling boot text over the handy new splash screen. Nostalgia aside, if you're experiencing errors on boot, the text can be a handy way to see at what stage in the process errors are occurring. There are two ways to return to the former flurry of text.

The first is to open your Raspberry Configuration by going to Menu > Preferences > Raspberry Pi Configuration and then selecting "Boot to CLI". Bear in mind that this will boot the Pi to the command prompt. To make the desktop and rest of the GUI appear, simply type the following command:

```
startx
```

The second way to see the old scrolling text is to run the following command to open the boot parameters:

```
sudo nano /boot/cmdline.txt
```

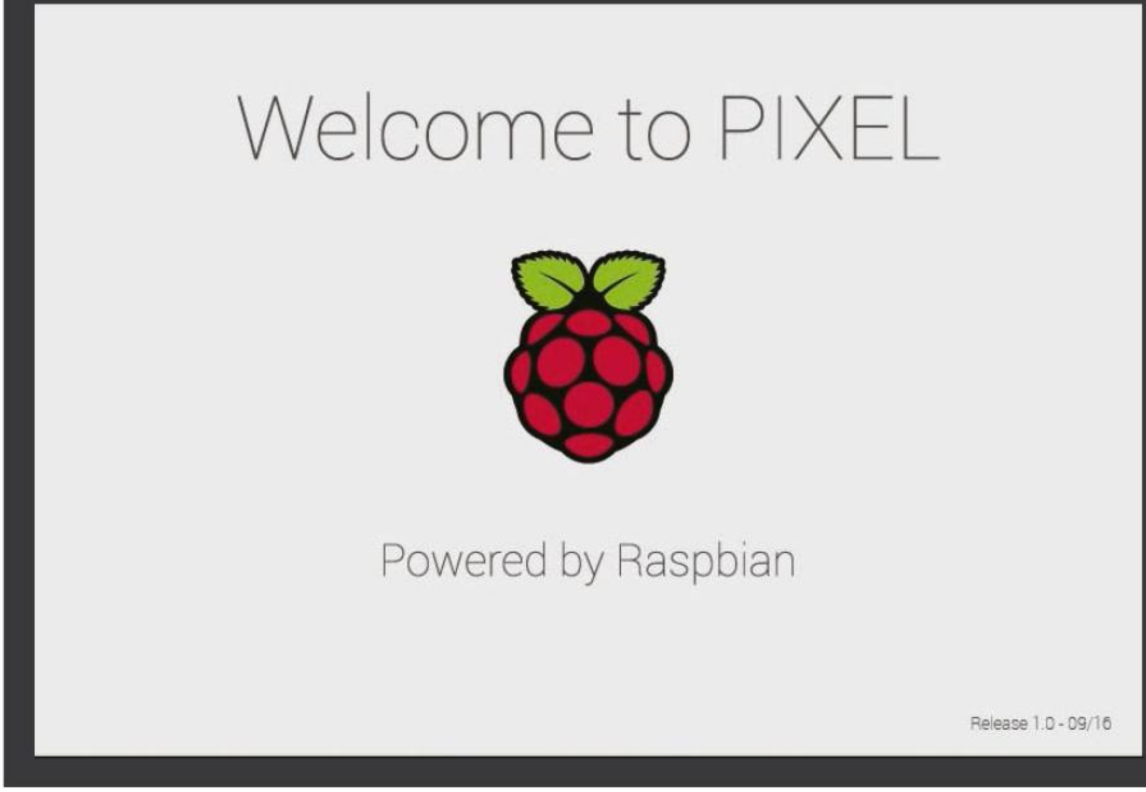
Now locate and delete the words "quiet splash". Next simply save your changes and exit.

If you are fond of the splash screen but feel it's a little sterile on its own, you can replace it with a picture of your choice. The splash screen image is located at **/usr/share/plymouth/themes/pix/splash.png**. If you do choose to replace it, make sure the new file is also called "splash.png". If you choose to do this, use the command line – the folder in question is owned by the root user. Open Terminal or connect via SSH and run the following command:

```
sudo mv /path/to/yournewpicture.png /usr/share/plymouth/themes/pix/splash.png
```

In case you want to revert in the future, make a copy of the old splash image named "old-splash.png" by running the command

```
sudo cp /usr/share/plymouth/themes/pix/splash.png /usr/share/plymouth/themes/pix/old-splash.png
```



❮ **The Pi Foundation claim that the splash screen doesn't slow down the boot process but if you prefer the old boot screen, it can be disabled.**

GET THE MOST FROM YOUR RASPBERRY Pi

OUT NOW!
WITH
FREE
DIGITAL
EDITION

The best Raspberry Pi guide money can buy!

148 pages of advice

Raspberry Pi
MADE SIMPLE

Learn how to hack, code and make with the incredible Pi

Includes NEW guides to Raspberry Pi 3

FREE
Digital edition of this book!
See page 146 for more information
iPad & iPhone only

Future

TECHNOLOGY MADE SIMPLE SERIES

Future

ORDER YOUR COPY TODAY!

Order online at **www.myfavouritemagazines.com**

Also available in all good newsagents

Available on the App Store

GET THE MAGAZINE ON Google play

The top 20 Raspberry Pi add-ons

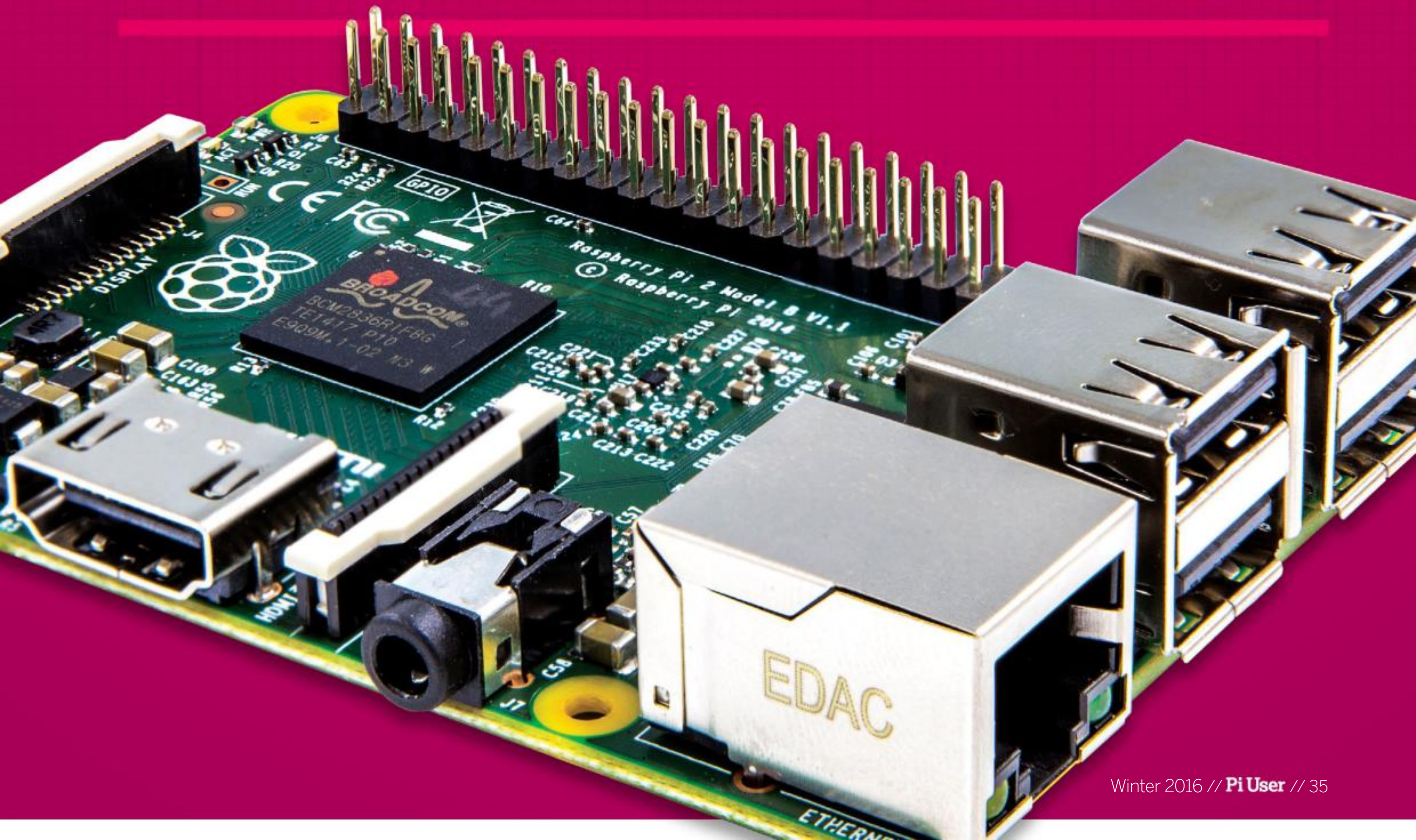
Add multiple dimensions to the way you use your Raspberry Pi with this collection of essential add-ons compiled by **Nick Peers**

The Raspberry Pi was designed to inject much-needed enthusiasm back into computing and electronics, and thousands of Pi users have showcased amazing and innovative projects that show just how versatile it can be. Part of this success is down to the Pi's fundamental design, with its USB ports and GPIO pins encouraging people to interact directly with the Pi to achieve… well, just about anything.

Many Pi projects have morphed into something more permanent as crowd-funding campaigns jostle with major manufacturers to provide people with all manner of add-ons to extend their Pi's capabilities, without having to design something from scratch. In this roundup we've focussed on 20 of our favourite Pi add-ons, all of which enable you to use your Pi in different ways.

You'll find out how to put your Pi at the heart of an all-in-one desktop or tablet, give it the tools to interact with its surroundings through the addition of sensors and robotic controls, plus power it via batteries or even your network. You'll discover some novel ways of using the camera module, including spying on your fish tank!

We'll reveal how to add Wi-Fi and extra USB ports to your Pi Zero if you need them, and you'll also find out how best to enjoy games on your Pi, whether it's through a simple Pi-compatible gamepad or by housing your Pi in a full-blown arcade cabinet. And the beauty of this roundup is, we've barely had time to scratch the surface – so if you find an add-on here doesn't quite do what you want, you'll almost certainly be able to find something that does elsewhere. And if you can't find it? Perhaps now is the time to build it yourself – and then share it with others. Isn't that what the Pi is all about?



pi-topPROTO
pi-topPROTO
pi-topCEED

1

pi-top CEED

Turn your Pi into an all-in-one desktop PC

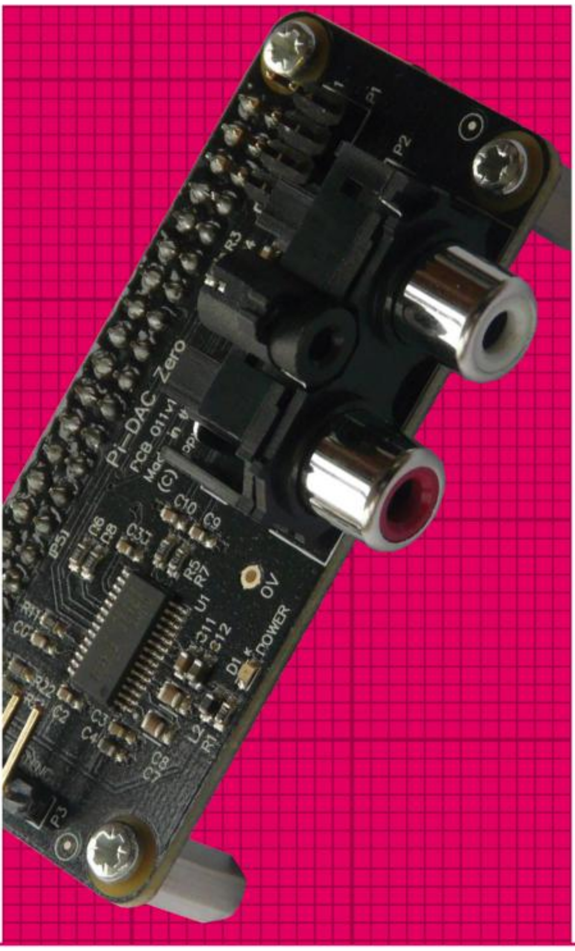
£119, http://uk.rs-online.com
This beautifully designed desktop case with integrated display will house any Raspberry Pi Model B form factor. The 14-inch display offers a decent 1366x768 resolution – equivalent to an HD-ready 720p screen – and the Pi itself hides behind the removable black acrylic panel beneath. All you need to supply is the Pi (or you can buy a CEED with Pi 3 included for £142), plus a mouse and keyboard.

Assembly is simple: remove the panel, attach your Pi to the mag rail and slide it in. Internal connections are neatly hidden away; all the cables you need are supplied, plus a power supply, and there's space for your HATs too. Your Pi's USB and Ethernet connectors are easily accessible; the adjustable kick stand allows you to position it for comfort.

There are two optional extras you can buy: the pi-topPROTO is a HAT-compatible add-on board for prototyping electronics; the pi-topSPEAKER is a tiny speaker you can also fit to the mag rail. You can attach up to three for stereo sound, but each is £20.

The CEED comes loaded with the pi-topOS and CEEDuniverse, a multi-player online game that teaches you coding and circuit-building skills so you can make hardware to use with the game.

Also consider: The Pi-Top case converts your Pi into a fully functional laptop, with 13.3-inch screen and 10-hour battery. Around £225, **www.pi-top.com**



2

Pi-DAC Zero

Turn a Pi Zero into an audiophile-friendly music player with this powerful DAC

£13 (fitted), www.iqaudio.co.uk
Give your old stereo system a new lease of life by pairing it with a Pi Zero. Just solder GPIO pins on to the Zero, then plug in the Pi-DAC Zero. This superb DAC supports full HD audio through its twin phono connectors – use a standard set of phono plugs to pair it with the AUX IN input on your stereo and you're away. Drivers are built into Raspbian and comprehensive instructions for enabling and using the DAC can be found at **www.iqaudio.com/downloads/IQaudIO.pdf**. Or you can follow the tutorial at **www.techradar.com/1318771** to pair your Pi Zero with the superb PiMusicBox distro to create a fully functional audio player you can control remotely from any web browser.

Rip your music in FLAC format for maximum quality and then sit back and prepare to be gob-smacked. Don't believe us? We've been using it together with PiMusicBox as our primary music player for a year, leaving the CD player to gather dust.

Also consider: You can add a headphone socket via the £12 Pi-DAC ZeroHP board, which plugs into the top of the Pi-DAC Zero. There's a full-size version of the Pi-DAC Zero (Pi-DAC+, £31) for Pi Models B+, 2 and 3, and a Pi-DigiAMP+ to turn a Pi into a full-blown stereo receiver.

3

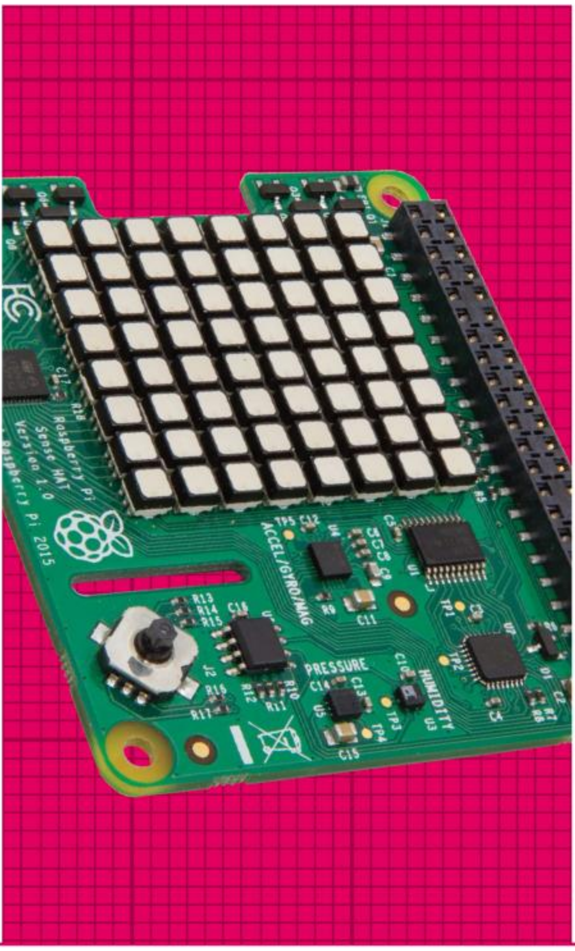
ZeroView

Take pictures with this superb Pi Zero case

£7, www.thepihut.com
You've bought a Pi camera, but what's the best way to mount it? Our favourite camera-themed case is the ZeroView, a cleverly designed window/glass mount for your Pi Zero (or Pi Model A+) and the camera module (don't forget the £4 Pi Zero Camera Adapter cable when you buy). Powerful suction cups secure it to any glass surface, and the camera module is close enough to practically eliminate glare and reflections. The way the Pi Zero is mounted means you can add additional pHATs, while all the connectors point downward for easy access (and tidier cables).

It's easy to assemble – no soldering involved – and the PCB board slides off the suction cups to make them easier to remove in turn should you want to shift your vantage point. Program your Pi Zero to record your fish swimming around its aquarium, or perhaps use the simple time-lapse command to shoot some stunning footage.

Pair it with MotionEyeOS (**https://github.com/ccrisan/motioneyeos**) to turn your Pi Zero into an IP camera so you can keep an eye on your home in real time from anywhere.
Also consider: For home security the SPI-BOX (£13, **www.sbcomponents.co.uk**) has a built-in motion detector. The Pi Camera Box Bundle (£22, **www.modmypi.com**) fits the module and Pi Zero in a wall-mounted case with wide-angle lens.



4

Sense HAT

Multi-purpose add-on board used in space

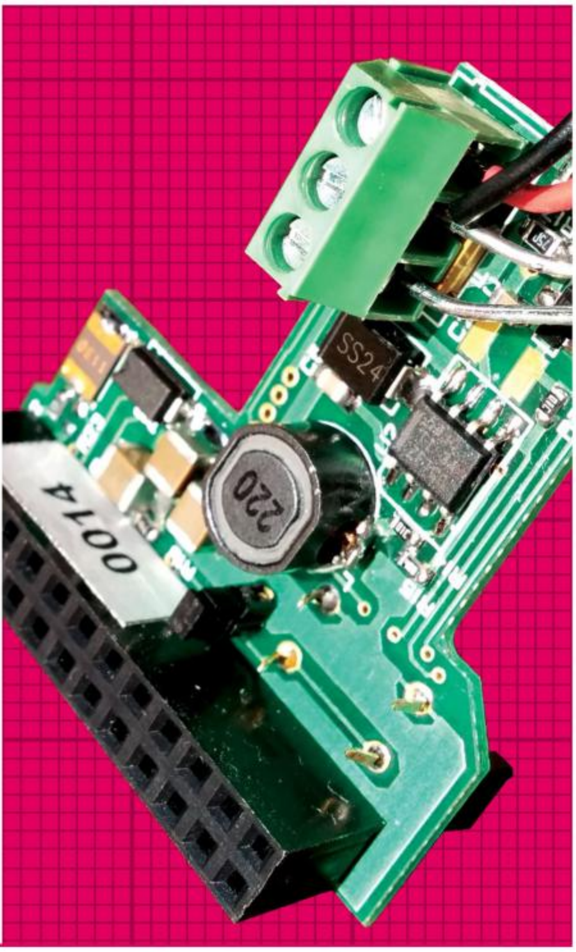
£29, https://cpc.farnell.com
The Sense HAT was designed to take the Pi to the International Space Station with astronaut Tim Peake – see **https://astro-pi.org**. It's also packed full of useful tools. The "Sense" refers to its six sensors, for measuring temperature, barometric pressure and humidity, plus gyroscope, accelerometer and magnetometer functions. You also get input controls via a five-button joystick and can output to the 8x8 RGB LED matrix.

Its versatility makes the Sense HAT ideal for everything from conducting experiments to playing games, and you can get started with its huge Python library at **pythonhosted.org/sense-hat**. If you can't wait that long you can get a feel for the Sense HAT before you buy it by running an emulator in your web browser (**https://trinket.io/sense-hat** – note it doesn't currently work in Firefox). This is also a great way to try out your coding projects before you apply them to your Sense HAT; there's a desktop version available for offline use through Raspbian too.
Also consider: Sense HAT will work with a Pi Zero, but Enviro pHAT (£16, **www.pimoroni.com**) is a better fit if your interest is the sensors. Its four sensors measure temperature, pressure, light level, colour, three-axis motion, compass heading and analogue inputs. There's a large Python library you can import to interrogate it with.

5

MoPi

Power your Pi from any portable source



£25, www.pimoroni.com
The Pi's small size suggests it's perfect for portable use, but it needs a power supply. Any portable battery designed to charge a smartphone over USB can be used with the Pi, or you could look at Pi Supply's PiJuice Standard (£25, **www.pi-supply.com**). Or, for maximum flexibility, consider the MoPi. It plugs into the GPIO header, but requires only four pins and can be used together with other GPIO-connected add-ons if you choose the Stackable version. Wire it up to any suitable power source – one capable of at least 6.2V under load – and after a bit of configuration with the aid of the excellent **https://pi.gate.ac.uk/pages/mopi.html** website you're away. Two default profiles are built in, one for non-rechargeable batteries and another for eight NiMH AA rechargeables. (You'll need to supply your own battery holder – expect to pay £3-4 for one that holds eight batteries, and make sure it has a PP3 clip to plug into the connector supplied with the MoPi.)

How much charge you get from your batteries depends on their load and capacity: aim for at least 2,000mAh per battery, which should give you up to ten hours use between charges even with several connected add-ons. The MoPi also provides an on-off switch and can cleanly shut down your Pi when the battery level drops to critical, ensuring you don't mess anything up.

6

7-inch touch screen display

Turn your Pi into a tablet

£60, cpc.farnell.com
The Pi's official touchscreen has an 800x480 display and 10-point capacitive touchscreen; install a third-party on-screen keyboard app such as Florence and you can ditch the keyboard too. Power is supplied from your Pi through the connector, which plugs into the Pi's DSI display port – we recommend using the newer 5.1V/2.5A power adapter to ensure the Pi and screen get all the juice they need. Pair it with the MoPi (above) to provide power on the move.

You could turn your Pi into a compact desktop unit if you pair it with the SmartPi Touch Case (£18, **pimoroni.com**), which tucks your Pi away behind the screen and provides access to all key connections, including the camera and display cable. The case also comes with a LEGO-compatible camera case that can be attached to the main case – perfect for using the camera as a webcam.

Also consider: Want a higher-res display? ModMyPi offers the Wareshare 10.1-inch 1024x600 touchscreen with case for £90, but it requires cables to connect to a Pi's HDMI port (display) and one USB port (touch support).

7

LCD Control Case Bundle

A case with a 3.2-inch touchscreen

From £26, http://shop.sb-components.co.uk
Want an embedded display but not a full 7-inch touchscreen? This case (available in black, blue or transparent) for Pi Model B+, 2 or 3 has a 3.2-inch TFT touchscreen, which connects to your Pi via the GPIO port to provide touch and a 320x240 pixel display. There are also three microswitches at the bottom, which you can program as you see fit.

Your Pi fits snugly into the bottom of the case, which can even be wall mounted, making it a good choice if you're using a Pi in a remote control project and want an attractive but accessible screen for interacting with. The case also ensures that the micro switches are accessible through a series of tiny holes, which you can get at using something like a paperclip.

The Bundle also handily includes a power adapter and USB Wi-Fi adapter if you need them, plus an 8GB microSD card containing a modified version of Raspbian. This works "out of the box" with the touchscreen, meaning you simply pop it in, boot up and can immediately start using the screen.

8

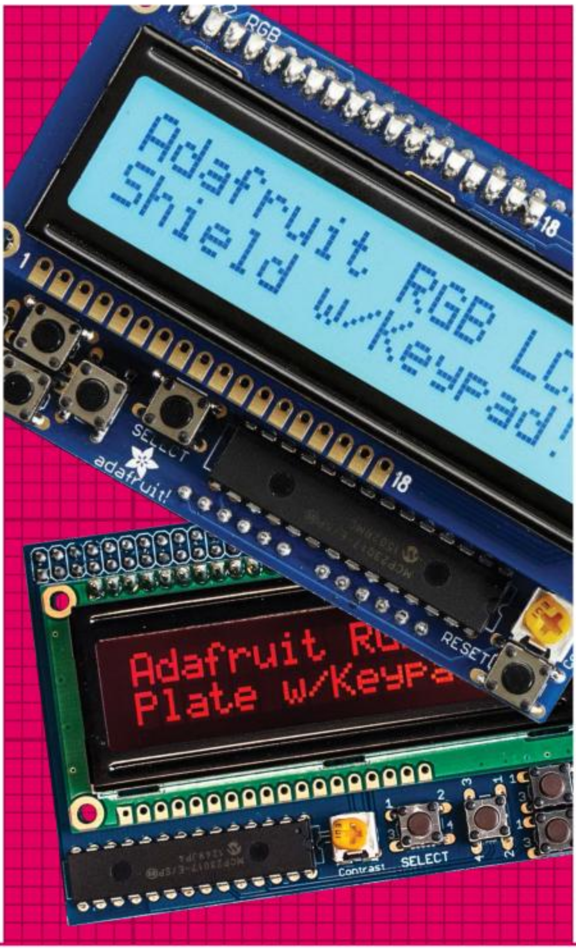
Adafruit 16x2 LCD & Keypad

Send simple messages from your Pi

£26, www.pimoroni.com
Adafruit's two-line LCD, available in positive and negative versions with RGB backlight, can display either a single message or one that scrolls to show longer ones.

The plate also has a four-way keypad and select button, so you can interact with the Pi using the standard I2C protocol (Adafruit includes several Python libraries you can tap into). The board plugs into the GPIO port to sit on top of your Pi for easy access – it uses just two pins, both of which can be shared with other I2C devices, leaving the other pins free for other uses. Note: if your Pi has four USB ports, you'll need to place electrical tape over these and any resistors to prevent direct contact with the LCD and Keypad.

Also consider: the Micro Dot pHAT (£22, **thepihut.com**) offers six LED matrices (red or green) plus a decimal point. It's meant for the Pi Zero but will work with any Pi. Even more basic, the Blinkt! (£5, **thepihut.com**) offers eight multi-colour RGB LED indicators that are individually controllable and dimmable, allowing them to flash on and off, pulse, etc. Useful to talk with your Pi in Morse code, maybe?



9

Pi Camera Module v2.1

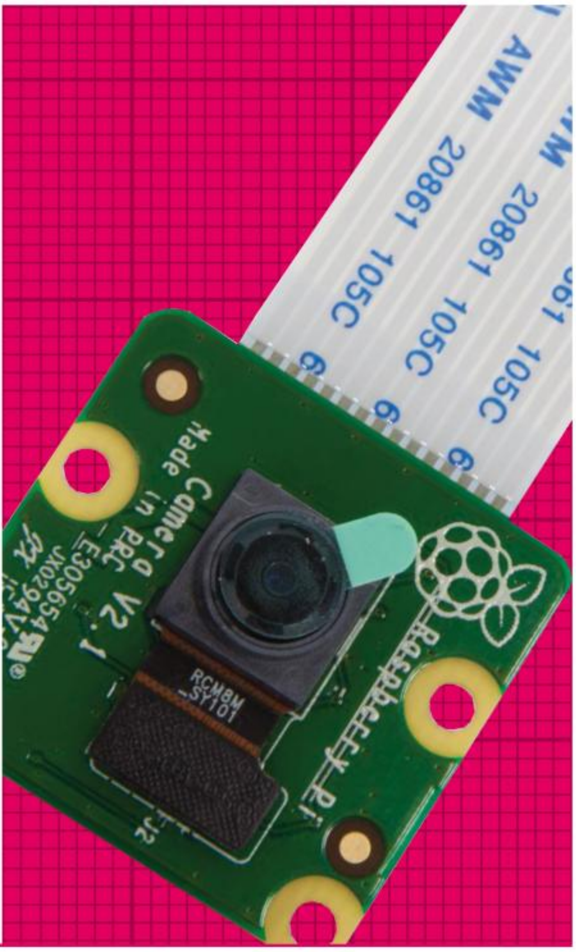
The official high-res camera add-on

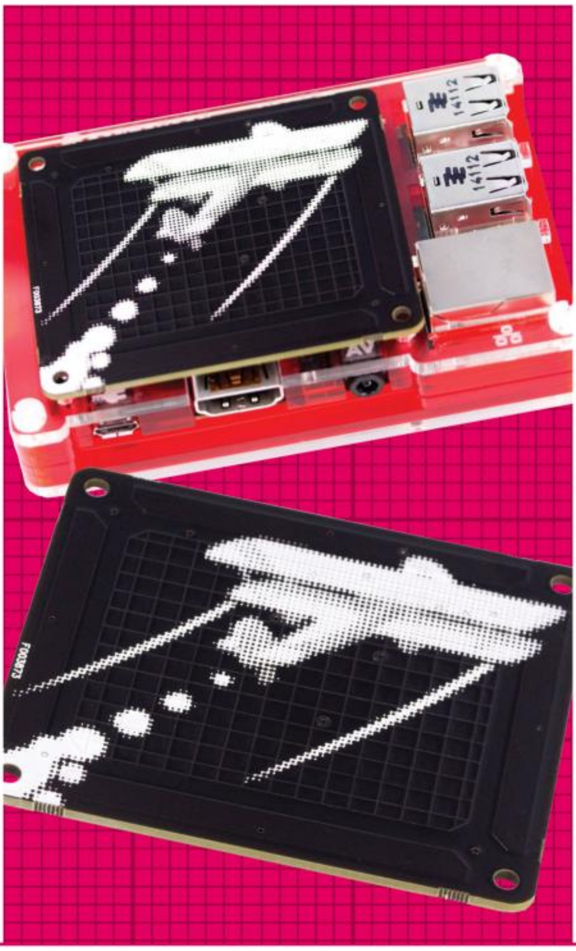
£21, http://cpc.farnell.com
The official Pi Camera Module packs an 8-megapixel Sony sensor that's also capable of shooting HD video, making it suitable for anything from time-lapse photography to webcam or home security use. Just add the right case and software.

There's one choice to make: the standard daytime version, or the NoIR model for use at night with suitable infra-red lighting? The camera connects to a special connector on the Pi via a ribbon cable – for Pi Zero, make sure it's the newer Pi Zero v1.3 with camera connector built-in, and get the special cable that plugs into the Pi Zero's smaller connector (around £4).

The camera is supported by an extensive Python library for programmers, and all you need to do is enable it via the Raspi-config utility.

Also consider: Pimoroni stocks a lens set (£8) with fish-eye, wide angle and macro lenses, complete with clip for attaching them to the Pi camera module. ModMyPi offers a mini camera stand with magnetic lens plate and a choice of lens (including telephoto and super fisheye) to securely clip on it for £9.





10

Skywriter HAT

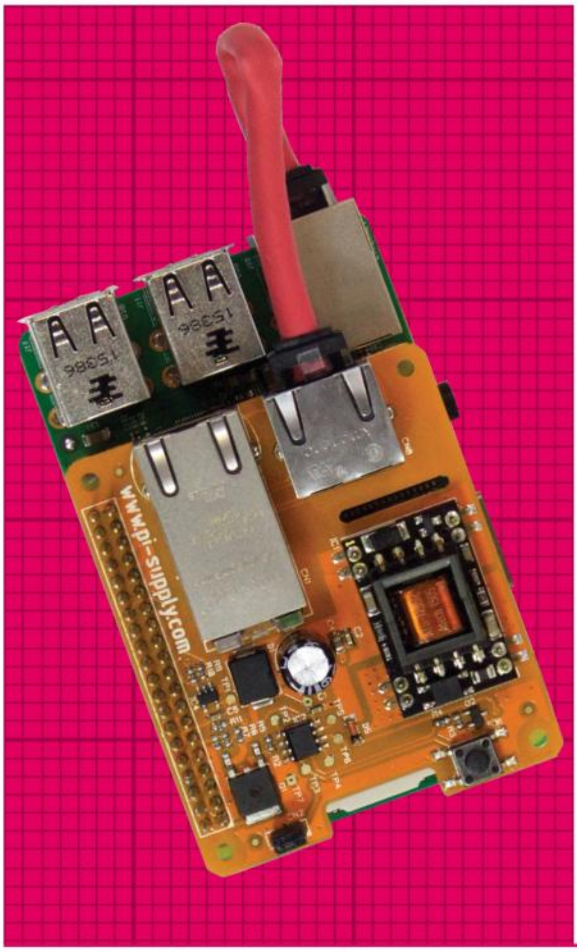
This clever add-on enables you to control your Pi using a series of gestures and taps

£16 www.pimoroni.com
Imagine holding your audience's attention with a wave of your hand or a flick of the wrist as you take control of your Pi. That's the magic behind the Skywriter HAT, which enables you to control your Pi by performing a series of hand gestures over the top of the HAT from up to 5cm away.

It works using an electrical near-field that circulates around the surface of the Skywriter in 3D. When you move your hand through this field, the HAT's Python API is able to measure the disruption and respond as programmed. It's capable of reading positional data in three dimensions along with common gestures such as flick left, tap and double-tap, giving you a wide range of control options.

So controlling your Pi using a wand while yelling "Expelliamus!" sounds cool, but what practical projects could you use the Skywriter HAT for? How about gesturing your way through a LibreOffice presentation? Or making sweet music using the synth.py script that's included as part of the HAT's Python API? You could even experiment with using it as a touchpad, although its lack of precision may leave you frustrated.

Also consider: Skywriter (£20, **pimoroni.com**). This is the big brother to the Skywriter HAT, and allows you to gesture up to 15cm away from its surface.



11

Pi PoE Switch HAT

Power your Pi from a network port

£30, www.pi-supply.com
Need power plus network access for controlling a project remotely or retrieving the data it collects? A PoE Switch HAT provides both power and a network connection from up to 100 metres away using a standard Ethernet cable. If you need to protect the unit, search for "Pi weatherproof case" to find some designs you can make using a 3D printer, or if the unit's staying in the dry, check out the PoE Switch HAT Case (£8, **www.modmypi.com**).

One more thing: you can't just plug into your router or switch and expect it to provide power. The switch needs to support the 802.3af PoE protocol, and most don't. In this case you'll need to add a PoE injector, which needs to sit next to your switch and a power socket. One is TP-Link's TL-POE150S (£20, PC World), which can provide up to 13W.

Also consider: The Pi PoE Switch Hat works with the Pi Model B+, 2 or 3 – if you have a Model B, then visit **www.xtronix.co.uk/raspberry-pi-poe.htm** for details of a compatible model. Pi Zero users will have to fashion their own solution – see **https://hackaday.io/project/9455** for one example.

12

Picade

Create your own arcade machine powered by your Raspberry Pi!

£180, www.pimoroni.com
The Raspberry Pi – in particular the Pi 2 or Pi 3 – is more than capable of running the most demanding games from decades gone by. Fire up the RetroPie gaming emulator and you can relive past gaming glories from various computers and consoles from the '80s, '90s and even the noughties. There's also support for MAME, which emulates thousands of old arcade games. All you need to do is supply your own game controller (see the final entry in this roundup).

But why not go the whole hog and emulate an arcade machine in miniature? Enter the Picade, a tabletop case with 8-inch screen (800x600 resolution, more than enough for retro gaming), stereo speakers, joystick and 12 micro-switched arcade buttons. It's a DIY job – allow up to three hours to put it all together, but there's no soldering involved and the result has an authentic feel thanks to the black powder-coated MDF and Perspex.

You also get the Picade PCB, which connects to the Pi via USB and converts the joystick and buttons (which are wired via the terminal contacts) into keyboard presses. It also provides a 3W amplifier for your speakers.

Also consider: Too pricey? £135 buys you the full cabinet without screen, £90 just the console part (including joystick and buttons), and £22 the Picade PCB.

13

Wi-Fi Adapter and USB hub

Add more USB ports and Wi-Fi to Pi Zero

£10, www.pimoroni.com
The one shortcoming of the Pi Zero is its single microUSB port. What if you want to use a mouse, keyboard and Wi-Fi? One answer is the Broadcom Wi-Fi Adapter and 2-port USB hub.

It plugs into your Pi's USB port – for Pi Zero you'll need a microUSB to USB cable or the £2 Shim you'll find at most Pi stockists. This makes it perfect for general use or just to get your Pi Zero up and running before you switch to headless use. The Y-shaped hub has a fast 150Mbps Wi-Fi adapter plus two full-size USB ports, and its low CPU load means little impact on your Pi Zero's performance. There's even a socket for an external power adapter – one with a 2.5mm jack providing 5V/2A power with a positive centre contact will do the trick.

Also consider: Want a discreet Wi-Fi adapter that plugs directly into the Pi Zero's lone microUSB port? The Pi Zero WiFi adapter (£8, **www.modmypi.com**) is a micro-sized USB dongle – no need for any messy adapters or shims. It's just 23mm long and requires no drivers but offers the same fast 150Mbps (802.11n) speeds as the Broadcom.

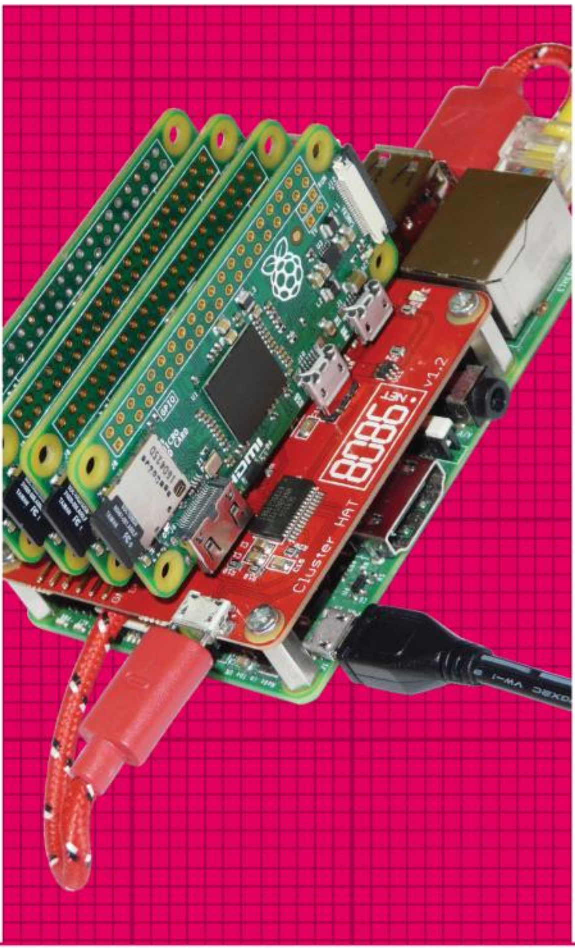
14

Cluster HAT

Discover how to combine the power of four Pi Zeroes with this clever little add-on

£28, www.clusterhat.com
Cluster computing harnesses the power of multiple computers to boost performance. The Cluster HAT enables you to plug in up to four Pi Zeroes to use them as an array. It requires the use of another Pi, which acts as the controller – the HAT plugs into this, and then you mount your Pi Zeroes on the top using USB. Power is supplied from the controller's GPIO, and you can control its output to each of the Pi Zeroes independently. You will, however, need microSD cards for each individual Pi Zero, on to which specialised Raspbian Jessie images will need to be installed from the Cluster HAT website (click the Software link).

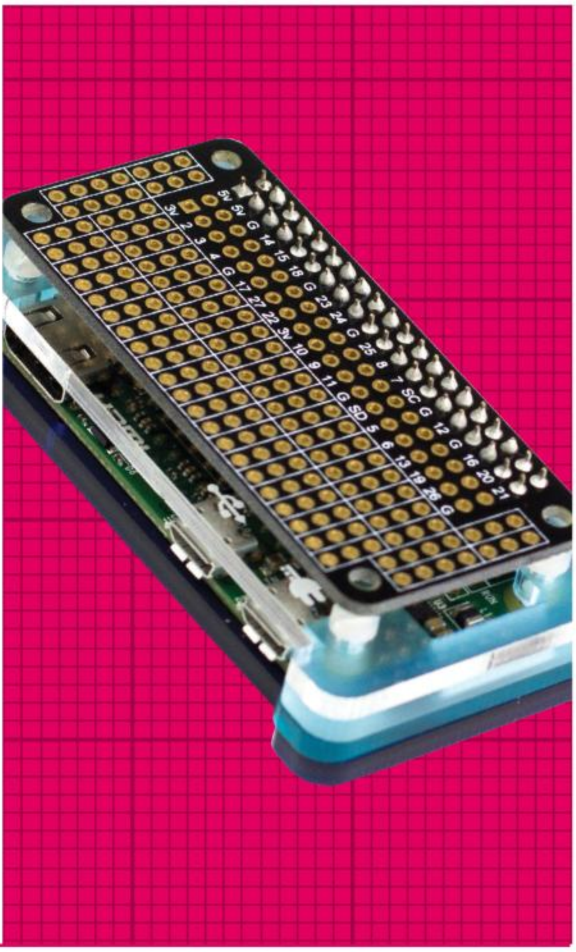
What you'll discover is that the four Pi Zeroes will work in tandem with each other on the same task, effectively combining their individual capabilities (including processor and RAM) to produce a faster computing experience. Climbers.net recorded a performance increase of over 400% when the four Pi Zeroes were combined using the Cluster HAT, although this is still less than half the performance of the Raspberry Pi 3 acting alone. Still, this is a cheap and effective way to start exploring the world of cluster computing, from teaching how it works to developing software that could then be ported to more powerful clusters.



15

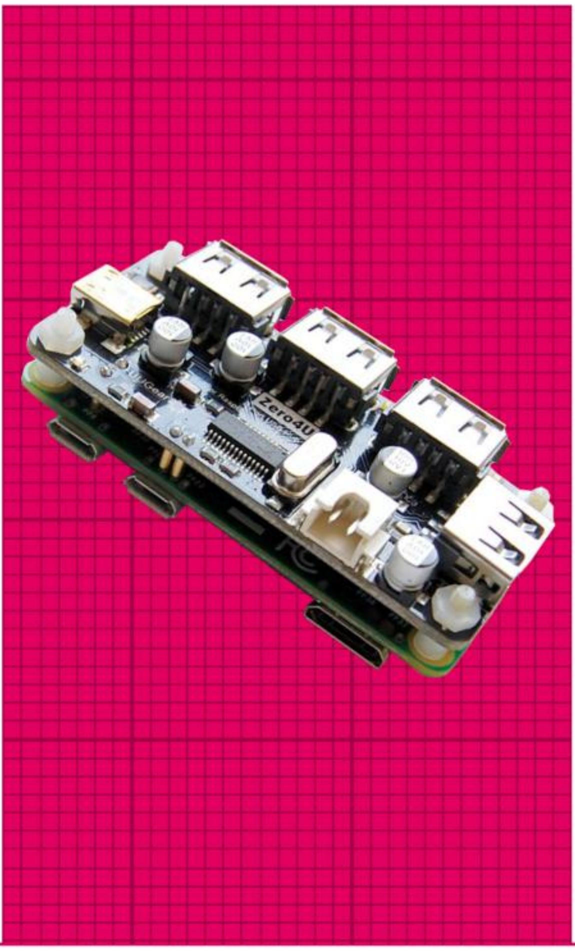
ProtoZero

Give your latest project a more permanent home by losing the breadboard



£5, www.pimoroni.com
The Pi's educational, hobbyist and practical side is never better expressed than in designing and building your own electronics projects. When it comes to testing your projects, you can't beat a good breadboard, which enables you to connect wires without soldering them, ensuring you don't make a costly or at least time-consuming mistake during prototyping. But at some point you'll want to move on from your breadboard – you'll either want it for another project, or its lack of portability will become annoying as your wires fall out for the umpteenth time.

The ProtoZero enables you to take your prototype to the next stage by giving you a board on which you can solder your components permanently. It's designed to match the Pi Zero's form factor, and sits neatly on top of the Pi Zero's GPIO pins (a female GPIO header is included, but not fitted). Here you'll also find two rows of corresponding – and clearly labelled – connector holes for the GPIO pins. There's also another 154 holes, split into logical lanes of three, into which you can solder your components on either side of the board, so you can hide wires. For more, visit **www.protoboards.co.uk**.
Also consider: The ProtoPal (£5, **www.pimoroni.com**) is the ProtoZero's big brother, and the shape and size of a regular HAT.



16

Zero4U

Add four full-size USB ports to your Pi Zero

£6.40, www.uugear.com
We've already looked at one way around the Pi Zero's chronic lack of ports: the Broadcom Wi-Fi Adapter and USB Hub. If you want something that better integrates with the Pi Zero, or you need even more USB ports, then the Zero4U is your lifeline.

It matches the Pi Zero's form factor exactly, but is cleverly designed to sit underneath it, so the GPIO pins and other ports are left accessible (and usable) at the top. Connecting the Zero4U to the Pi Zero requires no soldering – instead, there are four pogo pins on the back that line up with the +5V, GND, USB D+ and USB D- testing pad on the back of your Pi. Plastic screws are supplied to secure it, and the four full-size USB 2.0 ports each sport a handy white LED activity indicator.

It's configured to provide up to 2,000mA of current across the four USB ports, which is drawn from your power supply – if you want to connect the Zero4U to a regular Pi model, then you'll need a USB-mini USB cable to interface the two.

Note, there are two versions of the hub available: one for the original (v1.2) Pi Zero, and the other for the newer v1.3 Pi Zero (with camera connector). Make sure you get the right one! UUGear also provides a Zero4U and Pi Zero case for just £3.20.
Also consider: For larger Pi models, try the seven-port USB hub (£15.40, **www.uugear.com**).

DRUM HAT
DRH08
PIMORONI.COM
Framboisedorf
INSTRUMENT
OCTAVE
OCTAVE
C
D
E
F
G
A
B
C
Messrs E. Tjielman & P. Howard
Tokyo, New York, Barking, Norwich

17

DrumHAT & PianoHAT

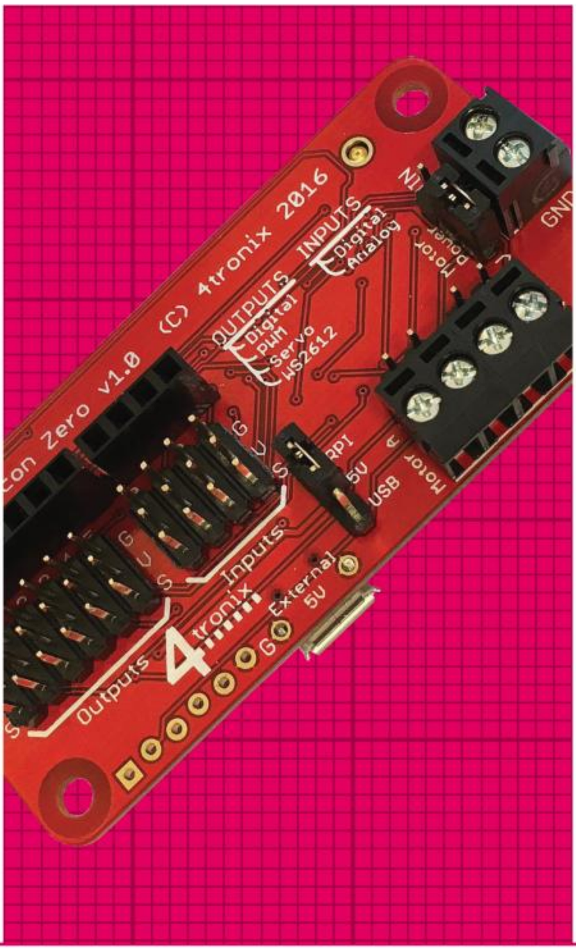
Turn your Pi into a music machine

Piano HAT £15, Drum HAT £12, www.pimoroni.com

The Piano HAT has 16 capacitive touch buttons, 13 arranged as a single-octave keyboard, plus controls for moving up and down the octave range and switching synthesised instrument. All 16 have LEDs you can program to light up when the key is pressed. A beginner's guide is on the Pimoroni website, plus a MIDI control option and a Python library, useful for running synths like Yoshimi, SunVox, playing music with Python or inputting sheet music via MuseScore.

The Drum HAT has eight capacitive sensor pads, each labelled with a specific drum and containing an LED that lights up when the pad is tapped. You can assign a sample to each pad (musical instruments, drums, voices, whatever) or program them to call any functions you like when tapped or released.

Also consider: The Adafruit Capacitive Touch HAT (£15.50, **www.pimoroni.com**) has 12 capacitive touch sensors you can connect to any conductive object using alligator clips (not supplied) to respond to a touch.


Picon Zero v1.0 (C) 4tronix 2016
OUTPUTS
Digital
PWM
Servo
WS2812
INPUTS
Digital
Analog
Motor
Motor
RPI
5V
USB
External 5V
Inputs
Outputs
4tronix
GND

18

4tronix Picon Zero

A controller for your next robotics project

£13.30, www.4tronix.co.uk

This is an intelligent robotics controller, which means there's a built-in process to handle all the direct communication between a robot's input and output devices. It uses the I2C interface, leaving all the GPIO pins free, and comes with two H-Bridge motor drivers, giving you forward and reverse, plus variable speed controls. The four general purpose inputs support digital, analogue or a DS18B20 temperature sensor. Up to six outputs are provided, supporting digital, PWM, servo and WS2812 (Neopixels). Both input and outputs use standard three-pin GVS connectors, which plug into most devices. Power can come either from your Pi or via separate adapters, and a dedicated socket supports the optional HC-SR04 ultrasonic sensor, for avoiding obstacles.

There's an extensive Python library, with ScratchGPIO support also promised. It plugs into any Pi that has the now-standard 40-pin GPIO header.

Also consider: MotoZero (£10, **www.thepihut.com**) is great for motor control – you can connect up to four, and it's Pi Zero size. Unlike the Picon Zero, it requires some soldering during assembly.

19

OSMC PiDrive

Create a media centre with its own storage

$80, www.modmypi.com

Like to turn your Pi into a media centre, capable of streaming and playing both local and remotely stored video, music and photos? The perfect operating system to use is OSMC (**www.osmc.tv**), a specially designed version of renowned media centre Kodi (**www.kodi.tv**) optimised for the Pi, and this PiDrive Kit should appeal to anyone looking for a single place to both store and stream their media.

The OSMC PiDrive Kit is a case designed to house both your Pi and the supplied 314GB Western Digital PiDrive, on which you'll store all your media files. The PiDrive has been designed to work more efficiently with the Pi than standard drives, drawing less power than other WD drives. It's based on the WD Blue slim drive, a 7mm high 2.5-inch drive. The case comes with a 3A power supply to power both your Pi and the drive from a single plug socket, OSMC preloaded on an 8GB microSD card, mounting media and a single power/data cable that connects the Pi and drive, as well as linking them to the 3A power supply.

Also consider: The 314GB drive is available separately for under £30 (**http://wdlabs.wd.com**), or £70 buys you the newer 1TB WD PiDrive Kit if you need additional storage. Note, this is 9.5mm high and comes with the WD PiDrive Enclosure and stand for housing your Pi and the drive, plus all required cables.

20

Pi compatible USB Gamepad

Add a new dimension to your gaming

£7, www.modmypi.com

Not sold on the idea of building a full-blown arcade cabinet? Then how about this KXD-branded gamepad? It follows the classic design with two analogue mini joysticks and a traditional D-pad, which offers eight-way direction. It has the full gamut of trigger buttons, from the X, Y, A and B buttons on the top to the dual left and right shoulder buttons, plus Start and Select buttons, giving you plenty of options during your gaming.

The gamepad should work with any emulator – it's plug and play with RetroPie – and a 1.4m cable means you don't have to hunch up next to the screen in order to immerse yourself in a gaming experience guaranteed to take you on a nostalgia trip.

One final point: if you already have a Windows analogue controller, you might find it's already compatible with RetroPie – plug it in and give a whirl first.

Also consider: Want to go even more old-school? Check out the SNES style USB gamepad (£4, **www.modmypi.com**). It has a traditional four-way D-pad, four trigger buttons and Start and Select, so will suit most games from the mid-'90s and before.

Future

lifehacker | UK

Helping you live better & work smarter

lifehacker | UK
SEARCH
FOLLOW
Future
LIFE
WORK
BUSINESS TECH
3 easy steps for faster PC
Start Now
1. Click Start Now
2. Run the quick scan
3. Fix the errors.
- Regclean Pro
PLANHACKER
FEATURED
ANNOUNCEMENTS
IOS
DOWNLOADS
3 easy steps for faster PC
MacBook Air

Planhacker: All the Best Samsung Galaxy Note 4 UK Deals and Tariffs

By Dan Grabham on 29 Sep 2014 at 1:30PM

Samsung's latest flagship productivity phone includes some super cool new features - here's all the best Note 4 UK plans and tariffs so far.

LIFEHACKER UK IS THE EXPERT GUIDE FOR ANYONE LOOKING TO GET THINGS DONE

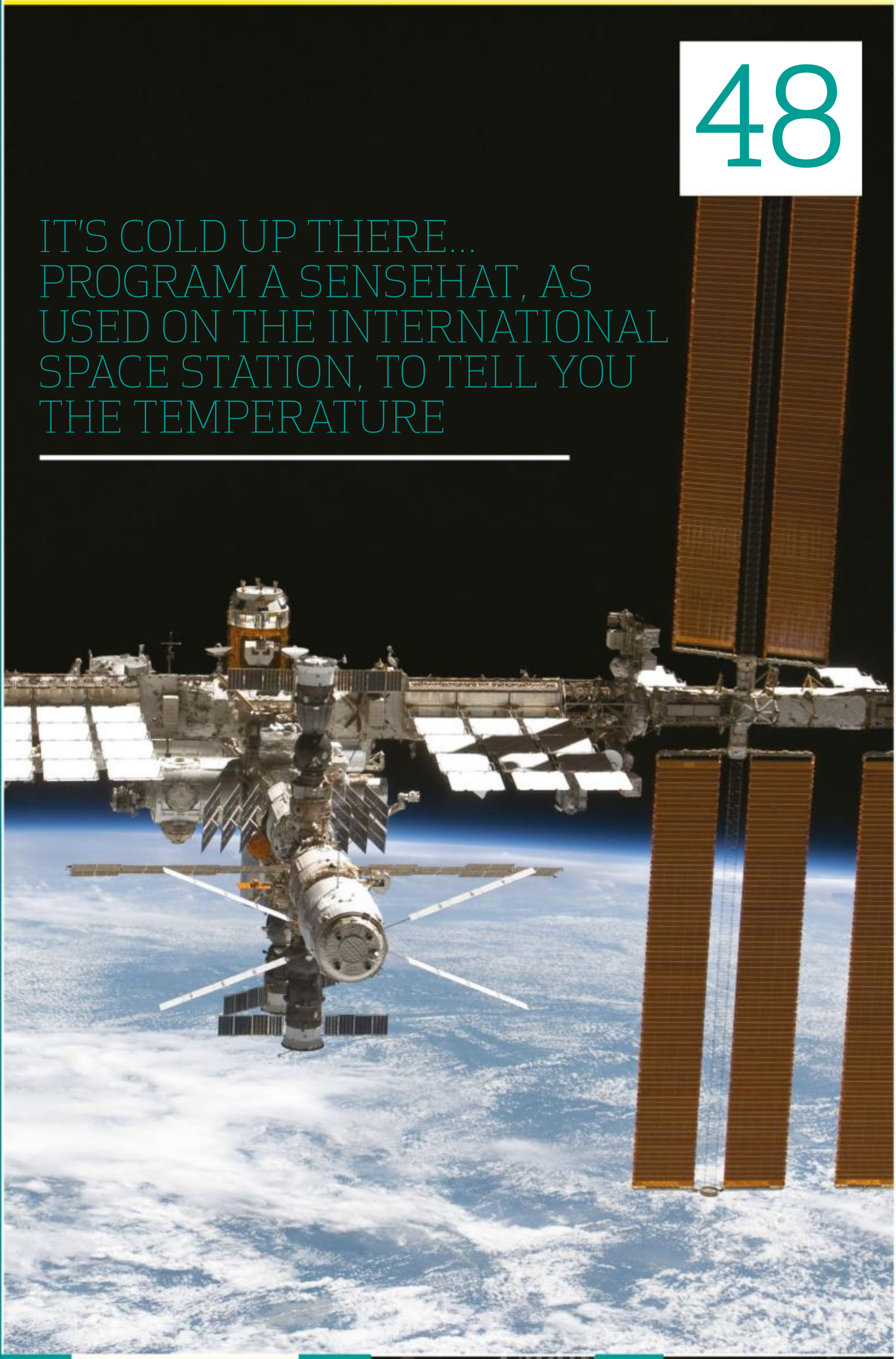
- Thousands of tips to improve your home & workplace
- Get more from your smartphone, tablet & computer
- Be more efficient and increase your productivity

www.lifehacker.co.uk

twitter.com/lifehackeruk facebook.com/lifehackeruk

Pi PROJECTS

Put your Pi to use – we show you how, step-by-step!



IT'S COLD UP THERE... PROGRAM A SENSEHAT, AS USED ON THE INTERNATIONAL SPACE STATION, TO TELL YOU THE TEMPERATURE

64

Create an admin account

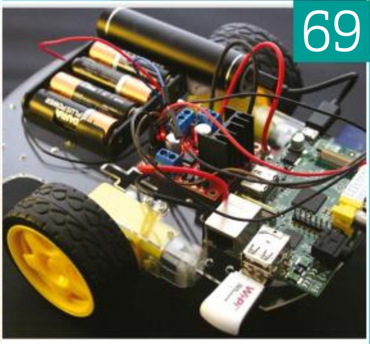
nextcloud1

●●●●●●●●●●

Weak password
Storage & database ▾

Data folder

/var/www/html/nextcloud/d



48 Get started with the SenseHAT

54 Recreate the Enigma machine

59 Build your own drone

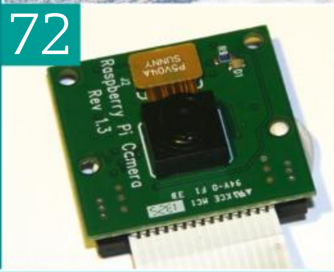
64 Set up a cloud data server

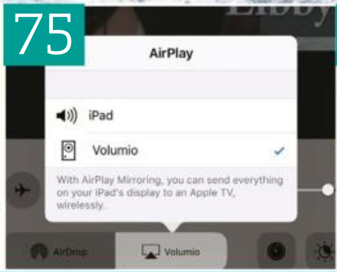
69 Build your first robot

72 Master the Pi Camera Module

75 Make speakers Wi-Fi-enabled

78 Enjoy internet radio on the Pi


72
Raspberry Pi Camera
Rev 1.3


75
AirPlay
iPad
Volumio ✓
AirDrop
Volumio

With AirPlay Mirroring, you can send everything on your iPad's display to an Apple TV, wirelessly.

78
6 CHNS Halifax
classic rock

❮ The SenseHAT was used in the Astro Pi project on the ISS.

Get started with the SenseHAT

Nate Drake introduces the sensational SenseHAT and explains how to use its sensors to measure the environment around you and execute commands

Although the Raspberry Pi is a powerful miniature computer, it's only as effective as the data feeding it. Many of the projects available for the Pi are only software-based, meaning they process inputs from the keyboard or mouse or data downloaded from the internet.

Using additional hardware, the Pi is capable of sensing a number of factors in your environment including the external temperature, the orientation of the Pi, light levels and much more.

There is virtually no limit to the applications for these. In this project, we will explore how to use the SenseHAT add-on for the Pi to sense the ambient temperature and alert you via email if it is too cold or too hot. This can be extremely useful for budding gardeners and we hope that once you become familiar with the SenseHAT and programming in Python (which we'll talk about shortly), you will take your coding further and create more exciting interfacing projects.

In order to proceed, you'll need a Raspberry Pi Model 2 or 3 with the latest version of Raspbian installed. You might also wish to invest in a SenseHAT, although as we'll see it is possible to get going without one...

Enter the SenseHAT

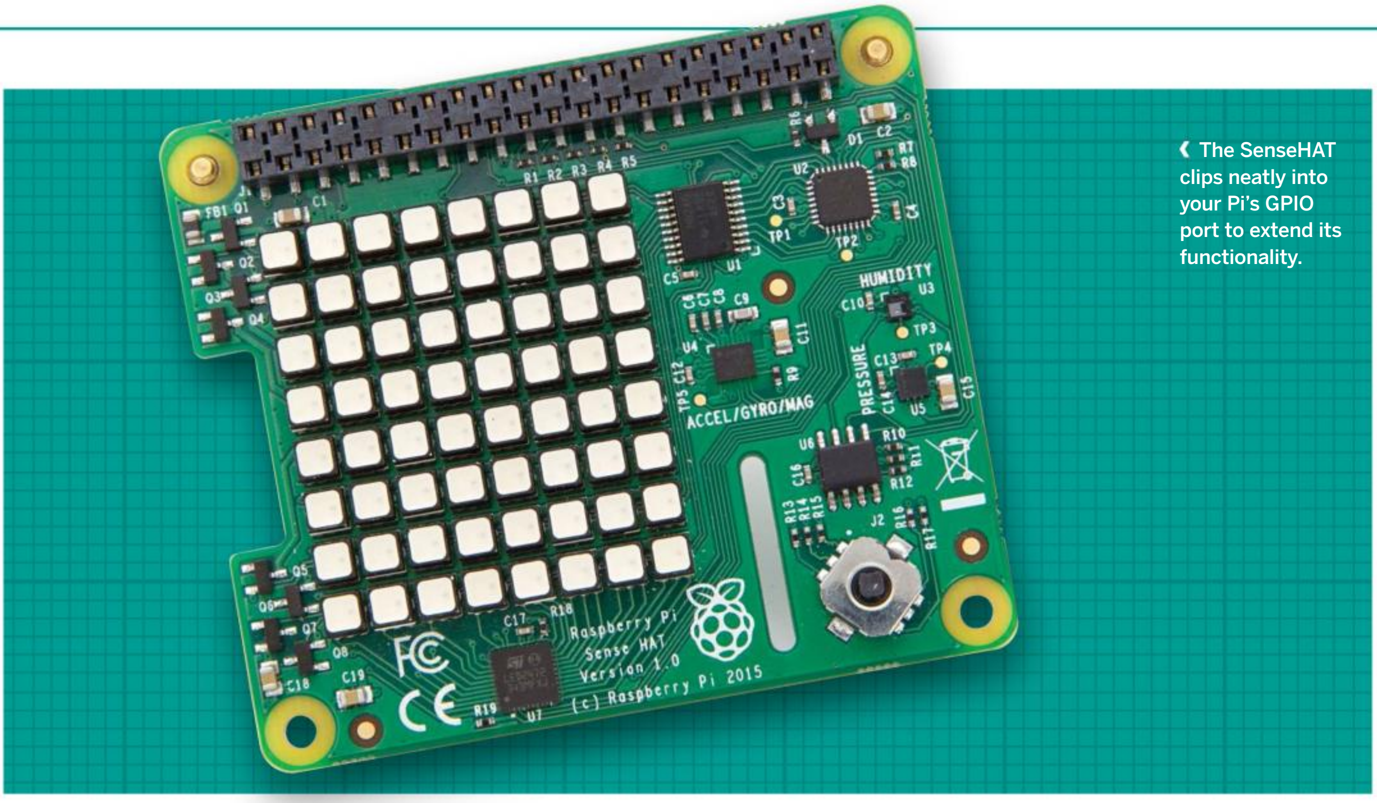
The SenseHAT is quite simply an add on board for the Raspberry Pi. HATs (Hardware Attached On Top) are hardware devices which, as the name suggests, can be connected easily onto the top of a Raspberry Pi via its GPIO (General Purpose Input/Output) connector. The HATs are made to an exacting standard and so most will work straight out of the box.

The SenseHAT was originally designed for the AstroPi Mission (more about this in a moment). It comes bristling with a veritable Swiss Army knife of gadgets and sensors, including a gyroscope, accelerometer, magnetometer, thermometer, barometer and humidity sensor. This makes it ideal for both budding astronauts and those interested in more mundane projects.

The SenseHAT also contains a 8x8 LED matrix display and a five-button joystick for interfacing projects. There are in fact two thermometers, one built into the humidity sensor and one built into the barometer. For the purposes of this project we will be using the default one built into the humidity sensor.

The SenseHAT is currently available from the Pi Hut website for £30 including shipping to mainland UK. See **https://thepihut.com/products/raspberry-pi-sense-hat-astro-pi** for more details. If you simply want to learn more about interfacing hardware and coding, the Pi also has a SenseHAT Emulator (more on this shortly).

Two AstroPi computers, which are quite simply Raspberry Pis with SenseHATs attached in a



The SenseHAT clips neatly into your Pi's GPIO port to extend its functionality.

special 3D-printed case, were flown to the International Space Station in December 2015. They formed part of British astronaut Tim Peake's mission, codenamed "Principia". They took advantage of the huge range of data the AstroPi can gather, and combined with the radiation and weightlessness of space allowed for experiments that cannot be performed on Earth.

Students from all across the UK took part in a competition and suggested various experiments to Tim, which he then performed inside the European Columbus module, making the results available for earth dwellers. See page 114 for more details.

To date only a small number of experiments have been run but fortunately the AstroPi's SenseHAT isn't useful just for investigating the question of life on Mars. Indeed, when taking your first steps with projects like these, you don't even need to buy any hardware beyond the Pi itself.

The amazing Emulator

The latest version of Raspbian comes with a SenseHAT emulator, which can be launched from Menu > Programming > SenseHAT Emulator. The emulator will provide input in exactly the same way as the real thing, except that you control elements like temperature, humidity and the joystick using on-screen slider bars and buttons.

Given the low cost of the SenseHAT, you may wonder what use the emulator is, considering

»

Pi bites

For a full list of the experiments run by Tim Peake and his colleagues on the ISS, see **https://principia.org.uk/the-mission/experiments-in-space/**

SOME LIKE IT HOT

Although the SenseHAT has an array of sensors, it's a good idea to start with a simple project, for instance to alert you if the temperature is too high or too low.

Go to Menu > Programming > Python 3 (IDLE) on the Pi. Now click File > New File to start a new project, then write in the following code:

```
from sense_hat import SenseHat
import os

sense = SenseHat()
tmin = 8
tmax = 27
temp = sense.get_temperature()
temp = int(temp)

if temp < tmin:
        print ("It's too cold")
        print (temp)
        # Uncomment the line below (remove #)
          to execute "alert" script.
        # os.system('python3 alert.py')

if temp > tmax:
        print ("It's too hot")
        print (temp)
        # Uncomment the line below (remove #)
          to execute "alert" script.
        # os.system('python3 alert.py')

if temp > tmin and temp < tmax:
        print ("Temperature is normal.")
        print (temp)
```

The values **tmin** and **tmax** are simply your chosen minimum and maximum temperatures for the greenhouse, so feel free to change these.

If you want to trigger an event – say, send an email in response to sensor data – then remove the **#** at the start of the lines reading **# os.system('python3 alert.py')**. Otherwise by default the program will run, display its message and then exit.

Go to File > Save and give the project a memorable name, such as **temp.py**. Make sure that both scripts are saved to your **/home/pi** folder. Once the project is saved, you can go to Run > Run Module from the Menu or press F5 at any time to execute.

It's time to start programming your SenseHAT using Python.

SET UP EMAIL ALERTS

Your alert script can execute any Python code. This sample script will send an email from a Gmail account to the email account of your choice, but feel free to experiment with Python to change the nature of the alert, for example by sending a Whatsapp message instead of an email.

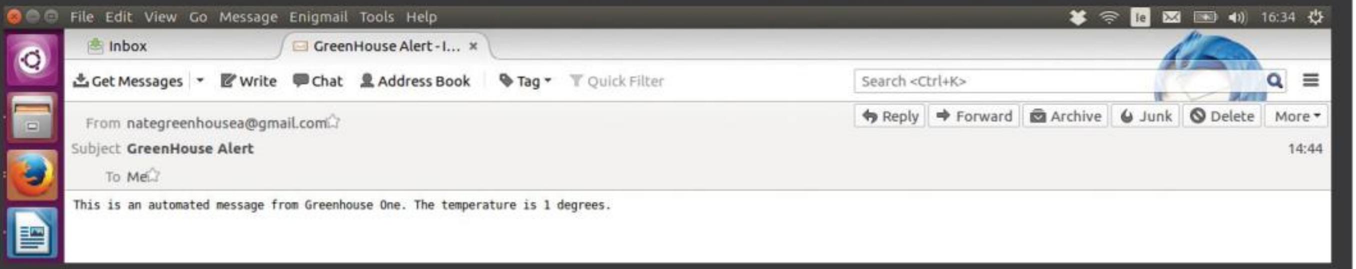
Go to Menu > Programming > Python 3 (IDLE) on your Raspberry Pi. Next click File > New File to start a new project. Paste in the following code:

```
from sense_hat import SenseHat
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

sense = SenseHat()

temp = sense.get_temperature()
temp = int(temp)
strtemp = str(temp)

fromaddr = "yourownemailaddress@gmail.com"
toaddr = "recepientemailaddress@gmx.com"
msg = MIMEMultipart()
msg['From'] = fromaddr
msg['To'] = toaddr
msg['Subject'] = "GreenHouse Alert"

body = "This is an automated message from Greenhouse One. The temperature is " + (strtemp) + " degrees."
msg.attach(MIMEText(body, 'plain'))

server = smtplib.SMTP('smtp.gmail.com', 587)
server.starttls()
server.login(fromaddr, "yourgmailpassword")
text = msg.as_string()
server.sendmail(fromaddr, toaddr, text)
server.quit()
```

Obviously change "yourownemailaddress@gmail.com" and "yourgmailpassword" to reflect your own details. For security, it's advisable to have a dedicated email account for this purpose. Also edit "recepientemailaddress@gmx.com" to the email address where you want to receive alerts. The default message simply contains an alert stating the current temperature at the time the email was sent; you can edit this too if you wish.

If you use a different email provider, you can amend "**server = smtplib.SMTP('smtp.gmail.com', 587)**" to reflect your provider's SMTP server and port – for example, for Yahoo, "**server = smtplib.SMTP('smtp.mail.yahoo.com', 465)**"

Go to File > Save and save the file as **python3 alert.py**. Make sure it is in the same **/home/pi** directory as the **temp.py** script you created on the previous page.



From nategreenhousea@gmail.com
Subject **GreenHouse Alert**
To Me

```
This is an automated message from Greenhouse One. The temperature is 1 degrees.
```

If you use email make sure to choose a distinctive name for your email alert account, like "Greenhouse A". This will avoid confusion later on.

that you can just plug in the real thing and execute your code. In the first instance, if you have very specific requirements and are not sure whether the SenseHAT is right for you, the emulator gives you a chance to "test drive" any code you've made. The emulator is also particularly useful when you test a number of different projects. It also allows easy sharing of any Python scripts you create through social media or over the web.

Any code you write for the emulator can be adapted for an actual SenseHAT by editing the first line of code (see "Sensory overload" on the opposite page for more information this). For the purposes of this project, use the Temperature slider to increase or decrease the temperature to test that your script works.

Pithy Python

The language used in the sample scripts is Python – specifically Python v 3.4.2. This distinction is important as different versions of Python can use different syntaxes and commands.

Python is the language of choice to interact with the SenseHAT and is one of many supported programming languages on the Pi through Python IDLE (Integrated Development and Learning Environment). This enables you to quickly generate and execute your own Python scripts. When using the Python IDLE program on the Pi, whenever you create a new project it will automatically be saved in your home folder. Make sure to save it with a meaningful name.

While Python is by no means the only programming language out there, it is arguably one of the easiest for beginners to master. There's also a huge amount of information available on the Raspberry Pi website and elsewhere to help get you started in Python.

There is no real limit to what Python can do, and you'll find that many Raspberry Pi projects have been built using this language. If you have specific requirements for your projects such as your Pi sending a tweet when your greenhouse becomes too warm, you can visit the main website's oddly named "Cheese Shop" (**http://pypi.python.org/pypi?%3Aaction=index**) to check that what you want is possible.

Even though it's very easy to learn Python terms, you should pay very careful attention to capitalisation of words and spacing as some Python scripts may not recognise commands without the proper lettering and/or indent.

Pi bites

The Emulator program has some helpful examples of code for Beginner, Intermediate and Advanced users. Simply go to **File > Open Example** to get started.

AstroPi VIS
LED Matrix
Joystick
DC IN
HDMI OUT
AV Terminal
LAN
USB
USB
UP
DN
L
R
A
B
IR Plate

The AstroPi's heavenly body. 3D blueprints for the casing are available on the Astro Pi Mission's website, https://astro-pi.org

Image credit: https://astro-pi.org/wp-content/

You might also see that some code online is marked as being suitable for Python 2 or Python 3. The latest version of Raspbian supports both as some commands are not available in the one version or another. Go to Menu > Programming to view the programs for each of these. The sample code for this project was written using Python 3.

Temperate timing

By default the supplied scripts are designed to run only once. This can be a problem if you want the Pi to continuously measure a value such as the ambient temperature.

The solution is the handy tool Cron, which is built into your Pi. It's used to schedule tasks and can be told to execute a script at certain times. For a full rundown of what Cron can do, we suggest reading the Raspberry Pi documentation at **https://www.raspberrypi.org/documentation/linux/usage/cron.md**. In brief, however, Cron is one way to have a script continuously check a value. The basic format of Cron is:

```
minute  hour  day of month  month  day of the
week  command
```

An asterisk must be used in place of any of these values that you are not using in a particular case. If for instance you wished to execute the script "myscript.py" in your Pi's home folder at midnight every night, the format would be:

```
0 0 * * *  python3 /home/pi/myscript.py
```

This is particularly handy if you're using email to send out alerts to avoid your mailbox being flooded with messages every few seconds.

In order to set up a Cron job, open Terminal on your Pi or connect via SSH and run the command:

```
crontab -e
```

Scroll to the very bottom of the text and paste your job (say the midnight script example we've just looked at). The first time you run the `crontab` command, you may be asked to choose an Editor. Choose Option 2 to use the simplest version.

Press Ctrl + X, then Y, then Return to save and exit. You can double-check scheduled jobs by running the command `crontab -l`

If you want your "sensor" script to run continuously, for example because you wish to use the LED display to show the temperature, then instead of using Cron you can simply modify the Python code. Simply add the following:

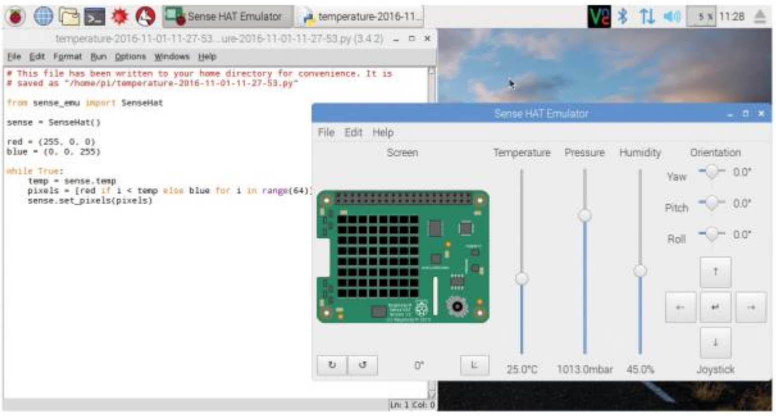
```
while True:
```

to the line directly above the first `if` to make the script run constantly.

You can find examples of code which will light LEDs in response to temperature in both the Emulator examples and in the SenseHAT documentation on the Raspberry Pi website – **https://www.raspberrypi.org/learning/getting-started-with-the-sense-hat/worksheet/**.

Sensory overload

The code you have seen in this tutorial is a very tiny slice of the Pi's potential to use Python and the SenseHAT, and should be seen as inspiration »



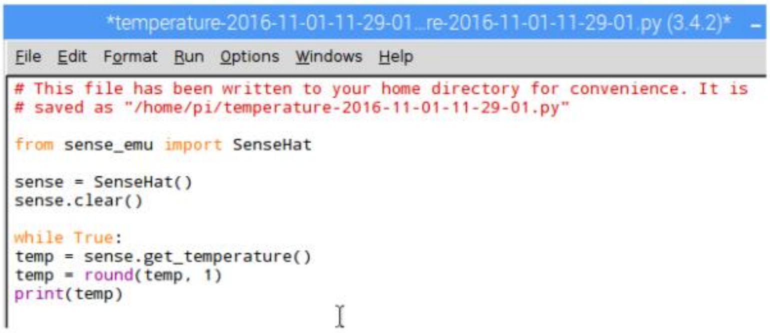
```
# This file has been written to your home directory for convenience. It is
# saved as "/home/pi/temperature-2016-11-01-11-27-53.py"

from sense_emu import SenseHat

sense = SenseHat()

red = (255, 0, 0)
blue = (0, 0, 255)

while True:
    temp = sense.temp
    pixels = [red if i < temp else blue for i in range(64)
    sense.set_pixels(pixels)
```

Sense HAT Emulator
temperature-2016-11...
temperature-2016-11-01-11-27-53...ure-2016-11-01-11-27-53.py (3.4.2)
File Edit Format Run Options Windows Help
Sense HAT Emulator
File Edit Help
Screen
Temperature
Pressure
Humidity
Orientation
Yaw 0.0°
Pitch 0.0°
Roll 0.0°
0°
25.0°C
1013.0mbar
45.0%
Joystick
Ln: 1 Col: 0
11:28

The emulator enables you to adjust the environment using buttons and slider switches. The sample code pictured is to change the SenseHAT's LED colours in line with temperature.

Pi bites

The Python website has some excellent tutorials for first-timers at **http://wiki.python.org/moin/BeginnersGuide/Programmers**


temperature-2016-11-01-11-29-01...re-2016-11-01-11-29-01.py (3.4.2)
File Edit Format Run Options Windows Help

```
# This file has been written to your home directory for convenience. It is
# saved as "/home/pi/temperature-2016-11-01-11-29-01.py"

from sense_emu import SenseHat

sense = SenseHat()
sense.clear()

while True:
temp = sense.get_temperature()
temp = round(temp, 1)
print(temp)
```

A simple Python script using While True:. Here the temperature is constantly displayed, rounded off to the nearest decimal place.

for more projects rather than an end in itself. The sample code is quite simple to execute if you have already installed a SenseHAT. If you choose to amend it or you simply wish to test pilot it before buying a SenseHAT, you can amend the scripts to use the Pi's built in SenseHAT Emulator. Simply go to Menu > Programming > Python 3 (IDLE) and change the very first line

```
from sense_hat import SenseHat
```

to

```
from sense_emu import SenseHat
```

You can then open the Emulator from Menu > Programming > SenseHAT Emulator and then use the slider switch to test different temperatures, or run your amended code to verify that everything works as expected.

Taking it further

As we previously discussed, the SenseHAT has its own gyroscope, humidity sensor, pressure sensor, joystick, LED display and much more. In particular for a greenhouse project, measuring the humidity as well as the ambient temperature can be useful for knowing when conditions are optimal for your plants.

If you're interested in understanding the SenseHAT further, the Raspberry Pi Foundation has produced its own free e-book "SenseHAT Essentials" (**https://raspberrypi.org/magpi-issues/Essentials_SenseHAT_v1.pdf**). If the coding examples provided in the emulator seem a little too basic for your tastes, you may also wish to head over to the Emulator developer's website (**https://trinket.io/sense-hat**) for examples of everything from how to take things a step further with the sensors to playing the Pi at "Rock, Paper, Scissors". The website also has a handy online SenseHAT emulator.

If any of your new projects require using the joystick, first try to use it out-of-the-box. If however you believe it needs calibrating, then follow the steps as outlined in the SenseHAT documentation at **https://www.raspberrypi.org/documentation/hardware/sense-hat/**.

If your Python 3 program isn't reporting any errors but you aren't receiving any alerts, try running the "alert" script on its own, either through opening it within the program or by opening Terminal and running the following command from within your home directory:

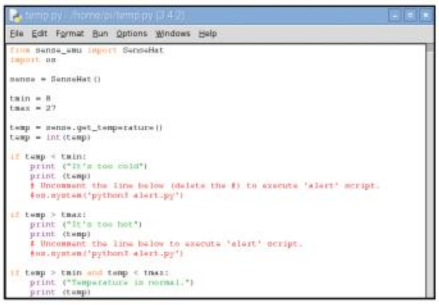
```
sudo python alert.py
```

This will help you narrow down the issue.

Another very helpful way to iron out any bugs in code without having to run the program every time from within the Python program is to have it check your "module" by holding Alt+X. This is especially handy if you've set up email alerts to save yourself from being bombarded by messages.

Finally, to follow the stellar AstroPi mission and see many more wonderful applications of the SenseHAT, be sure to visit the main website at **https://astro-pi.org**. ■

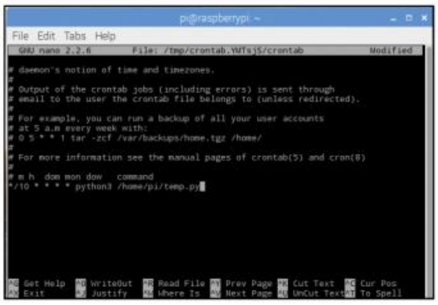
SETTING UP YOUR SENSEHAT JOB


temp.py - /home/pi/temp.py (3.4.2)
File Edit Format Run Options Windows Help

```
from sense_emu import SenseHat
import os

sense = SenseHat()

tmin = 8
tmax = 27

temp = sense.get_temperature()
temp = int(temp)

if temp < tmin:
    print ("It's too cold")
    print (temp)
    # Uncomment the line below (delete the #) to execute 'alert' script.
    #os.system('python3 alert.py')

if temp > tmax:
    print ("It's too hot")
    print (temp)
    # Uncomment the line below to execute 'alert' script.
    #os.system('python3 alert.py')

if temp > tmin and temp < tmax:
    print ("Temperature is normal.")
    print (temp)
```

1 Create sensor script

Build your sensor script as described in *Some Like It Hot* on page 49. Remember to uncomment the lines in the "alert" script if you want to trigger an action in response to sensor data. Save the script with an easily recognisable name – say, **temp.py**. into your **/home/pi** folder.



```
  GNU nano 2.2.6          File: /tmp/crontab.YWTsj5/crontab               Modified

# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/10 * * * * python3 /home/pi/temp.py
```

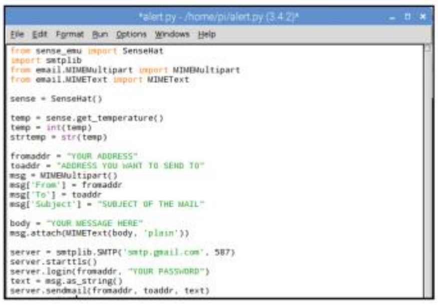
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

2 Schedule Cron job

By default the script you created will check the sensors only once. If you want it to do so regularly, you'll need to tell the Pi to execute it at certain times. Open Terminal on the Pi and run the command `crontab -e`. Scroll to the bottom of the file and add the following:

```
*/10 * * * * python3 /home/pi/temp.py
```

The Pi will now run **temp.py** every ten minutes.


alert.py - /home/pi/alert.py (3.4.2)
File Edit Format Run Options Windows Help

```
from sense_emu import SenseHat
import smtplib
from email.MIMEMultipart import MIMEMultipart
from email.MIMEText import MIMEText

sense = SenseHat()

temp = sense.get_temperature()
temp = int(temp)
strtemp = str(temp)

fromaddr = "YOUR ADDRESS"
toaddr = "ADDRESS YOU WANT TO SEND TO"
msg = MIMEMultipart()
msg['From'] = fromaddr
msg['To'] = toaddr
msg['Subject'] = "SUBJECT OF THE MAIL"

body = "YOUR MESSAGE HERE"
msg.attach(MIMEText(body, 'plain'))

server = smtplib.SMTP('smtp.gmail.com', 587)
server.starttls()
server.login(fromaddr, "YOUR PASSWORD")
text = msg.as_string()
server.sendmail(fromaddr, toaddr, text)
```

3 Create alert script

This step is optional but adds the icing on the cake. Follow the steps in *Set Up Email Alerts* on page 50 to create a script that will run if certain conditions are met – your greenhouse temperature falls above or below the limits you set up previously. Don't be afraid to customise this script in any way you see fit, but be sure to save it in the same folder as the main script you created in Step 1.

techradar.

The home of technology

techradar.com

Make an amazing Enigma machine

Nate Drake explores how to crack the code and set up an authentic version of the Enigma cipher machine used by the German military during WW2

Some people thought we were at war with the Germans. Incorrect. We were at war with the clock. These words were attributed to Alan Turing, as played by Benedict Cumberbatch in the 2014 film *The Imitation Game*. Turing was referring to the Enigma machine, which was being used by the Germans to encode military radio messages during World War 2 and to date had proved near impossible for the Allies to crack.

In technical terms Enigma machines were what's known as "electro-mechanical rotor cipher machines". They resembled large typewriters. An operator would configure the machine according to pre-agreed settings and encode a message using the keyboard. For each letter that was pressed, the mechanical parts would complete an electric circuit. The corresponding encoded letter would then light up on the machine. Each key press shuffles at least one rotor one place forward, meaning that the same letter is never encoded the same way twice.

A four-rotor Enigma machine. Different machines were used by different branches of the German military.

Image credit: Greg Goebel, Public Domain

Thanks to programmer Brian Neal's py-enigma library, it is now possible to reproduce the machine exactly using the programming language Python. The Raspberry Pi is perfect for running as an Enigma machine, thanks to the fact that it supports Python and has an excellent hardware random number generator. In this project we will explore how to download the py-enigma library to the Raspberry Pi as well as how to configure the Enigma machine in a way similar to the operators in World War II. We will also explore the workings of Enigma devices in order to understand how the Pi will encode your messages.

Enigmatic myths

The Enigma machine was invented sometime after World War I, although its inventor, German engineer Arthur Scherbius, initially saw only lacklustre interest from private companies.

With the outbreak of World War II the military of several countries started taking an interest in the machine and began adapting it by beefing up its security. So it's misleading to talk about "the Enigma machine" – there are several. Indeed, just as different countries adapted Enigma to their needs, different military departments within Germany used variations of the machine too.

The py-enigma library is capable of emulating the German Heer (Army) and Luftwaffe (Air Force) Enigma machines by default. It's also capable of emulating the Kriegsmarine (Naval) Enigma, which had a more sophisticated design. For the purposes of this project we are going to concentrate on the simplest implementation of Enigma used by the Army and Air Force.

Another common misconception about Enigma is that it was singlehandedly cracked by the code-breakers of Bletchley Park. It was in fact initially

ENIGMA ENCRYPTION

To encode messages you'll first need to navigate to where the handy enigma command line program is stored. Connect via SSH or open Terminal on the Pi and use the **cd** command to do this – e.g.

```
$ cd /home/pi/bgneal-enigma-f3af458a5d2d
```

Next examine your key sheet and determine which day's settings you want to use. Decide on three random letters to use as the message key.

You'll next run a command to encrypt the message key. Leave the rotors in the same starting position as in your keyfile for now.

```
$ sudo python pyenigma.py --key-file
yourkeyfilename --start='ABC'
--day=DAYOFMONTH --text='MESSAGE KEY
TEXT'
```

For instance if it's the 29th day of the month, the initial rotor positions in your keyfile are APJ and you want to use the message key UVX then you would write:

```
$ sudo python pyenigma.py --key-file /home/pi/
keyfile.key --start='APJ' --day=29
--text='UVXUVX'.
```

This gives the text output "QMMZIY".

Next use the message key to change the starting position of the rotors and encrypt the actual message, for instance:

```
$ sudo python pyenigma.py --keyw-file /home/
pi/keyfile.key --start='UVX' --day=29
--text='MEET ME AT MIDNIGHT'
```

This gives the text output "HPZXUWHMICMHUFCXOTU".

Decryption works using the same procedure in reverse. First set Enigma to the day's key settings to obtain the message key:

```
$ sudo python pyenigma.py --key-file /home/pi/
keyfile.key --start='APJ' --day=29
--text='QMMZIY'
```

Next use the message key UVX to decrypt the actual message:

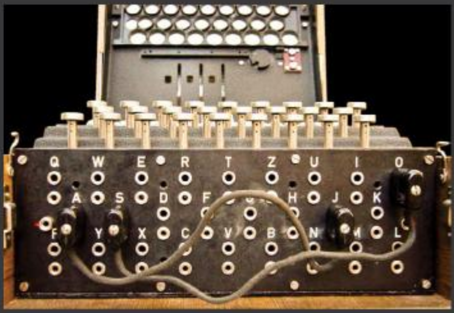


Image credit: Matt Crypto, CC BY-SA 3.0

Cables were used to connect letters on the plugboard. Here A is paired with O and Y with J. The standard number of pairs was ten.

```
$ sudo python pyenigma.py --key-file /home/pi/
keyfile.key --start='UVX' --day=29
--text='HPZXUWHMICMHUFCXOTU'
```

This decodes to the original message "Meet me at midnight".

broken by a Polish Cryptographer Marian Rejewski. Rejewski designed the first clockwork "bombe" machines which were used to test various Enigma settings and crack the code. This represented a huge leap forward as the Polish previously had had to resort to perforated cardboard sheets known as "Zygalski sheets" to try various permutations.

Alan Turing designed a number of more sophisticated versions of the Polish Cipher Bureau's bombe, which, as depicted in *The Imitation Game*, worked in combination with a "crib" – some known text in a message, such as the word "Wetter" (weather), which could be used to decode German messages.

The Enigma Machine used by the Army and Air Force consisted of several parts. The keyboard when pressed completed an electrical circuit which would light up the encoded letter. For example, when encrypting a message starting WET…, the operator would first press the W key, and the Z lamp might light up, so Z would be the first letter of the "ciphertext". The operator would then press E and note down the ciphertext letter, and so on.

Anatomy of Enigma

Each key press would turn at least one of the rotors once. The rotors formed the core of the powerful code behind Enigma. Operators would place three rotors out of a possible five on a spindle inside the machine. Each rotor had 26 electrical contact pins on one side to represent

Pi bites

A working replica of the Turing-Welchman Bombe is at the Bletchley Park Museum. For a video of it in action, see **http://bit.ly/Turing-bombe**.

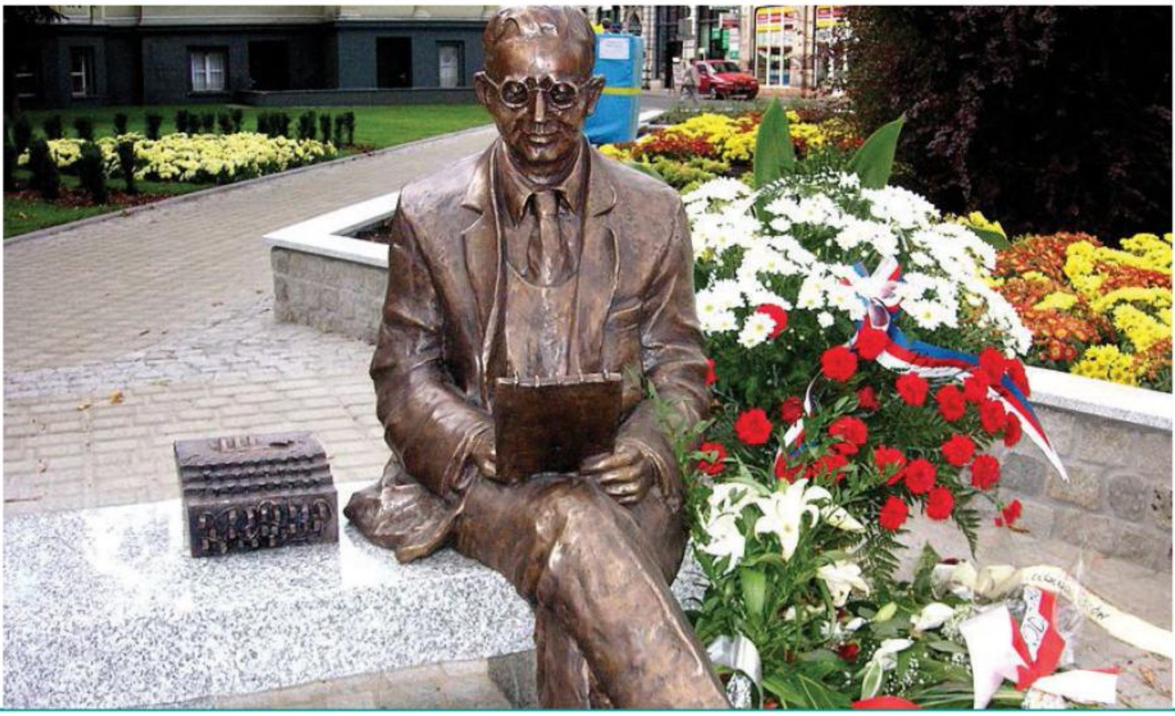


Image credit: Wojsyl, CC BY-SA 3.0

Monument to Marian Rejewski in Bydgoszcz, Poland. Note the ornamental Enigma machine next to him.

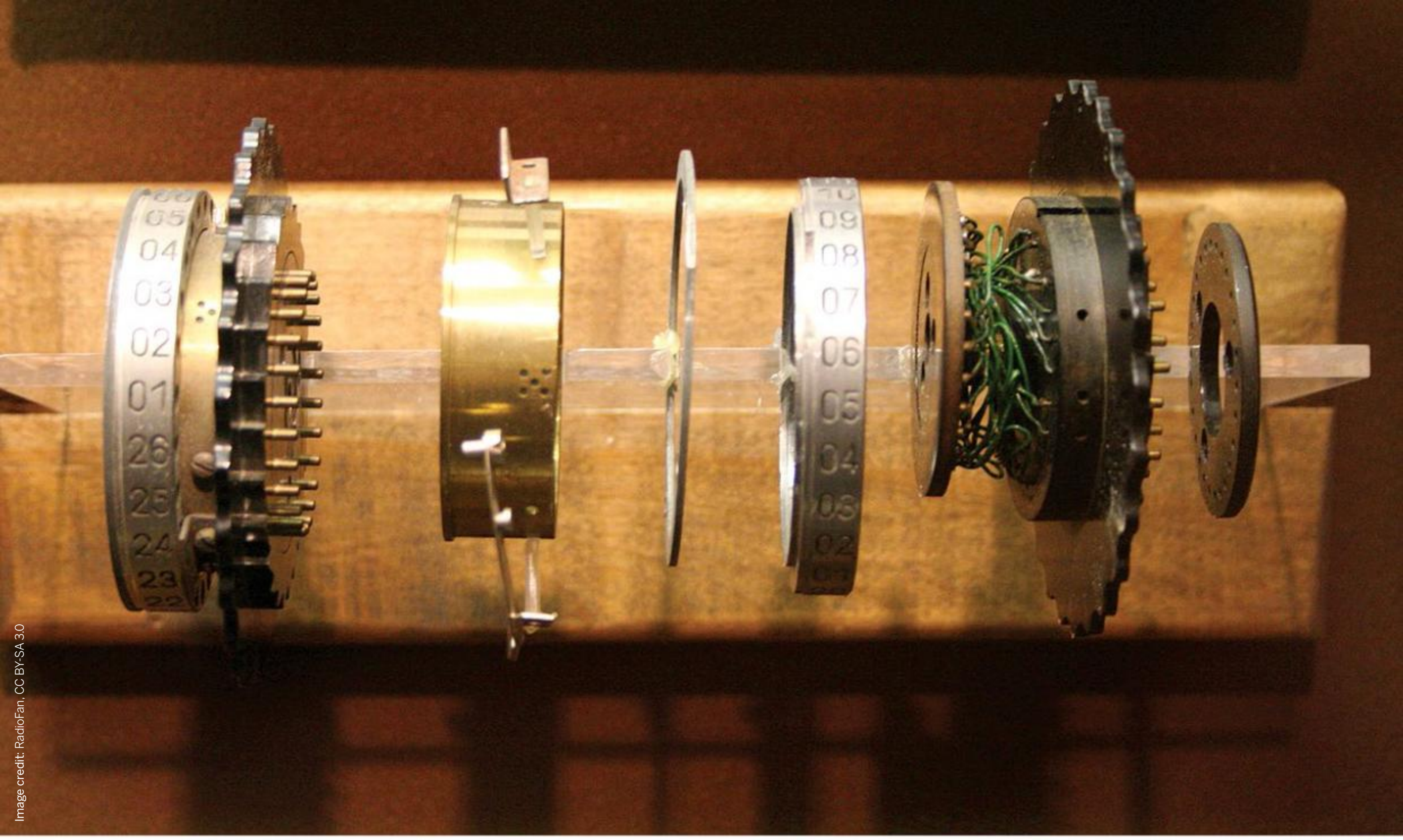


Image credit: RadioFan, CC BY-SA 3.0

Enigma rotor detail on display at the National Cryptology Museum. The internal wiring for each separate rotor, for example Rotor V, was the same from machine to machine.

» each of the letters of the alphabet, and the rotors were labelled using Roman numerals from I to V so they were easy to tell apart. When placed in an Enigma, each rotor can be set to one of 26 possible positions. This was known as the "Ringstellung" (literally the "ring setting") and allowed one letter to be encoded as another.

By itself a single rotor would offer little protection for your message. It would be set up in such a way that for example when you press the letter A, it would complete a circuit so that the letter T would light up on the Enigma. The letter B might be Q, and so on. This is a simple substitution alphabet and it could be broken very easily simply by examining the most common letters to work out, which are E, T, A and so on, until the rest of the message could be deciphered.

The cunning of Enigma was achieved through using multiple rotors. Each of the rotors had notches at different points along the alphabet ring. This resulted in a different substitution alphabet being used as the rotors turned. The rotor on the extreme right would move one letter forward each time a key was pressed and the others less frequently, depending on the rotor settings. Provided that a message was short enough, the word "Deutschland" for instance could be encoded any number of ways in the same message, making it near impossible to crack.

Allied attempts at breaking the code were frustrated even further by the "Steckerbrett" (plugboard), which was positioned at the front of the machine below the keys. Letters were wired together in pairs through cables before the text was encoded. For instance the letter F could be paired with the letter J. Having "steckered" pairs of letters such as these would mean that each time the letter F was pressed, the signal would be diverted to the letter J before going through the rotors, which would hugely increase the encryption strength of the message.

One final feature of the Enigma was the "Umkehrwalze" (reflector). The reflector could not be easily changed but as the name suggested made sure that the encryption and decryption process were the same. This meant that two operators who had set up their Enigmas with the same settings could easily encode and decode each other's messages. One drawback of the reflector is that a letter could never be encoded as itself, which is a weakness the Allies exploited.

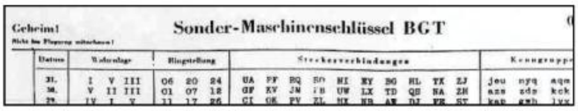
Pi bites

Programmer Louise Dade's website has an excellent key book generator which will create a PDF of random key settings for up to 40 days. See **http://enigma.louisedade.co.uk/dailykeys.html**

Enigma operators

Enigma operators were handed a "key sheet" by an officer containing the Enigma settings for that day. First the choice and positioning of the rotors would be listed – for example, IV, V and I. Next would be the initial position of the rotors either as letters or as numbers – say "A F P" or "1 6 16". The plugboard settings would follow. The Enigma machine allowed for up to 13 pairings, but the Army standard was 10.

Finally the "Kenngruppen" would be listed. These were a series of letters that allowed operators to identify on which particular day the message was sent. This is one area where the py-enigma library on the Pi diverges from the original Enigma machine as its key sheets simply place the numbered day of the month at the start. See *The Key to Enigma* opposite for more.



Geheim!

Sonder-Maschinenschlüssel BGT

Image credit : Tony Sale, http://www.codesandciphers.org.uk

Example of a daily key sheet. For security reasons, officers would often tear a strip off the sheet to give operators the settings for that day only.

THE KEY TO ENIGMA

Py-enigma closely matches the settings for the original Enigma machine but adds the option to choose your own Umkehrwalze (reflector). These were hardwired into the original machines and usually weren't changed in the field. For more information on what the reflector is and does, see **http://users.telenet.be/d.rijmenants/en/enigmatech.htm#reflector**. For now we suggest using type B, which was the one used by the German Army.

The easiest way to store key settings is through use of a key file. Open Terminal on your Pi or connect via SSH and type:

```
sudo nano keyfile.key
```

This will create a blank file. You write the settings in the format Day, columns, rotor settings, letter pairings and reflector.

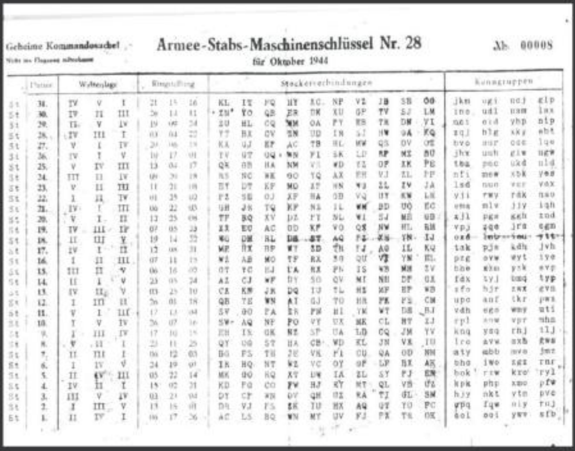
Paste in the text below by way of an example to get started:

```
# My sample key settings file
# First number represents the day of the month
29 II IV V A P J AV BS CG DL FU HZ IN KM OW RX B
28 IV II I C S M FA GX HL IQ KZ ME NS PR TB WC B
27 II IV V F V P PK QH RV SA UJ WO XC ZB DL GM B
```

You can adapt the keyfile to choose different settings and/or different Enigma machines if you like. For more information see **http://py-enigma.readthedocs.io/en/latest/keyfile.html**.

Once you've pasted in the settings you need, save and exit by pressing Ctrl + X, Y, then Return.

Make sure that everyone with whom you want to exchange messages has a copy of the keyfile.



Geheime Kommandosache!

Armee-Stabs-Maschinenschlüssel Nr. 28

für Oktober 1944

Copy of the Keysheet used by the Germany Army in October 1944. Py-enigma doesn't use the "Kenngruppen", so you can ignore these if you wish.

The Germans quickly realised that if thousands of messages were sent every day using the same settings then eventually the rotors would all revolve fully and encrypt messages in the same way, making them vulnerable to being cracked. For this reason, an operator would decide on a "message key" for each message.

First, the Enigma machine would be set up exactly as outlined in the key sheet. The operator would then choose three letters at random – say, "DAX" – and transmit these to the receiver. The convention at the time was to repeat the message key twice, making our example "DAXDAX". The operator would then change the rotor positions on their Enigma to match the message key and send the message itself.

To decode a message, the receiving operator would firstly only have to input the encoded message key into Enigma to work out that their counterpart is using the message key "DAX". They could then change the positions of their rotors and decrypt the message itself.

Breaking Enigma

The reflector was a major Achilles heel for Enigma because no letter could be encoded as itself. This meant that if, for instance, you believed that the word "ENGLAND" was somewhere inside a message, then you could eliminate any group of letters containing those letters.

The Allies also made use of "cribs" – where at least some of the message was known – to crack the day key. For instance the Germans usually transmitted a weather report around 06:00 daily which followed a predictable format and would contain the word "WETTER" (weather).

Certain features chosen by the Germans to increase security actually weakened Enigma. Rotor positions were not allowed to be the same two days running, meaning that if the day key had previously been cracked then a whole range of possible rotor positions could be eliminated. The same held true for plugboard settings.

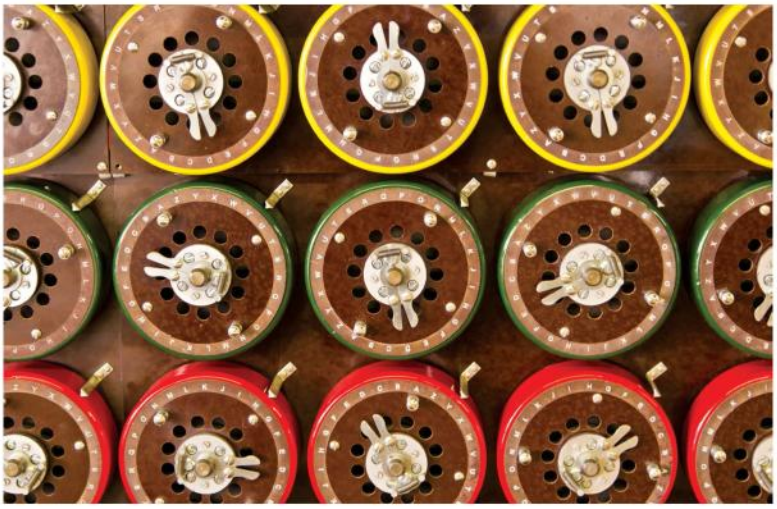


Image credit: Antoine Taveneaux, CC BY-SA 3.0.

A replica of the Turing-Welchman Bombe at the Bletchley Park Museum. The bombe could try thousands of rotor positions per hour, which made cracking Enigma much easier than doing this by hand.

Despite the wisdom of using a message key, German operators would sometimes be lazy and repeat keys or use ones grouped together on a keyboard. These were termed "cillies", a name that may have come from the predictable day key "CIL".

Aware of these shortcomings, the German Kriegsmarine increased the strength of Enigma by adding three more rotors, making a total of eight. They also avoided sending predictable messages, so the German Naval Enigma remained largely impenetrable throughout the war.

Submarine commanders were also given Enigma code books printed with red ink upon paper that dissolved readily in water, an added precaution to prevent seizure.

Ian Fleming, who gathered information for his James Bond novels while stationed at Bletchley, »

› The complexity of the Enigma machine was more than equalled by the complexity of the ingenious mechanisms devised to crack the code.

ventured a plan to capture Naval Enigma books which would do even 007 proud. His suggestion was to disguise some British commandos as downed German airmen and have them float around in the middle of the Channel on a captured fighter plane, looking helpless. Fleming believed a German U-boat would inevitably surface to assist their supposed countrymen, at which point the commandos would pop the hatch and raid the sub. The plan was shelved indefinitely for being too risky but does give some insight into how valuable Enigma intelligence was.

Thanks to the Official Secrets Act, the incredible achievements of the codebreakers of Bletchley Park went unrecognised until the '70s. The British Government took advantage of this after World War II by selling Enigma machines to Britain's former overseas colonies and happily decoded their messages for decades.

Enigmatic issues

As the py-enigma software can easily be run from the command line, there are few ways that this project can go wrong. As we've noted, there were several different types of Enigma machines, so if you want to exchange messages, it's important that you decide to use the same kind. The military M3 (three-rotor) machine is perhaps the most well known and easiest to use.

While this is an excellent project to do for fun, the fact that the Allies broke Enigma during World War II should be reason enough not to rely on it to store any truly private information.

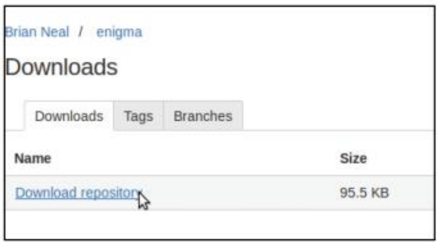
For a more in-depth history of how the Allies cracked Enigma, feel free to visit the Bletchley Park website at **www.bletchleypark.org.uk/content/hist/worldwartwo/enigma.rhtm**. For more information on py-enigma specifically, or if you wish to congratulate Brian on his masterpiece of coding, visit **http://py-enigma.readthedocs.io/en/latest/**.

Brian has also not rested on his laurels but has taken the time to compile a Python library for two other cipher machines. The first is the M-209, which is a cipher machine employed by the US military during World War II. To download the library see htt**ps://bitbucket.org/bgneal/m209/**. Brian has also created a library for the PURPLE Machine, which was used by the Japanese Foreign Office before and during World War II. See **https://bitbucket.org/bgneal/purple/** for more information. Enigma, M-209 and Purple can happily co-exist on the Raspberry Pi, so feel free to have fun with all three. ■

Pi bites

You can use the Pi to create a truly random message key with this command: sudo cat /dev/hwrng | tr -dc 'A-Z' | fold -w 3 | head -n 1

SET UP YOUR ENIGMA MACHINE



Brian Neal / enigma

Downloads

Name	Size
Download repository	95.5 KB

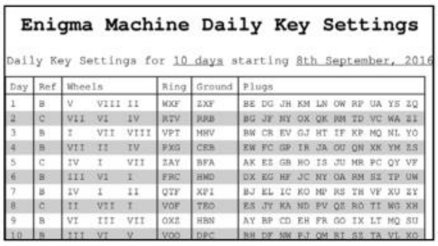
1 Download py-enigma library source files

Open Terminal or connect via SSH and run the following command to download the library:

```
$ wget https://bitbucket.org/bgneal/enigma/get/f3af458a5d2d.zip
```

Alternatively use your Pi's web browser to download the file.

Next unzip the file by running `$ unzip f3af458a5d2d.zip` then use the command `$ cd bgneal-enigma-f3af458a5d2d` to be taken into the newly extracted folder.



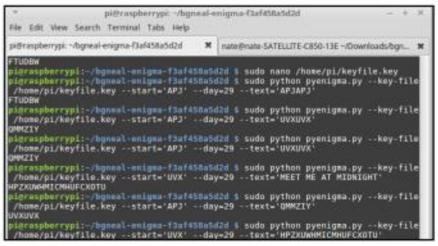
Enigma Machine Daily Key Settings

Daily Key Settings for 10 days starting 8th September, 2016

Day	Ref	Wheels			Ring	Ground	Plugs
1	B	V	VIII	II	WXF	ZXF	BE DG JH KM LN OW RP UA YS ZQ
2	C	VII	VI	IV	RTV	RRB	BG JF NY OX QK RM TD VC WA ZI
3	B	I	VII	VIII	VPT	MHV	BW CR EV GJ HT IF KP MQ NL YO
4	B	VII	II	IV	PXG	CEB	EW FC GP IR JA OU QN XK YM ZS
5	C	IV	I	VII	ZAY	BFA	AK EZ GB HO IS JU MR PC QY VF
6	B	III	VI	I	FRC	HWD	DX EG HF JC NY OA RM SZ TP UW
7	B	IV	I	II	QTF	XPI	BJ EL IC KO MP RS TH VF XU ZY
8	C	II	VII	I	VOF	TBO	ES JY KA ND PV QZ RO TI WG XH
9	B	VI	III	VII	OXZ	HBN	AY BP CD EH FR GO IX LT MQ SU
10	B	III	VI	V	VOO	DPC	BH DF NW EJ QM RI SZ TA VL XO

2 Prepare keyfiles

Use either a specialised key generator website or make up your own settings for the rotors and plugboard. This would also be a good time to decide on the kind of Enigma machine you want to mimic (three-rotor, four-rotor, etc.). Once you've generated enough keys, make sure to provide a paper copy to everyone with whom you want to exchange messages. See *The Key to Enigma* on the previous page for more information.



pi@raspberrypi: ~/bgneal-enigma-f3af458a5d2d
File Edit View Search Terminal Tabs Help
pi@raspberrypi: ~/bgneal-enigma-f3af458a5d2d
nate@nate-SATELLITE-C850-13E ~/Downloads/bgn...

```
FTUDBW
pi@raspberrypi:~/bgneal-enigma-f3af458a5d2d $ sudo nano /home/pi/keyfile.key
pi@raspberrypi:~/bgneal-enigma-f3af458a5d2d $ sudo python pyenigma.py --key-file
 /home/pi/keyfile.key --start='APJ' --day=29 --text='APJAPJ'
FTUDBW
pi@raspberrypi:~/bgneal-enigma-f3af458a5d2d $ sudo python pyenigma.py --key-file
 /home/pi/keyfile.key --start='APJ' --day=29 --text='UVXUVX'
QMMZIY
pi@raspberrypi:~/bgneal-enigma-f3af458a5d2d $ sudo python pyenigma.py --key-file
 /home/pi/keyfile.key --start='APJ' --day=29 --text='UVXUVX'
QMMZIY
pi@raspberrypi:~/bgneal-enigma-f3af458a5d2d $ sudo python pyenigma.py --key-file
 /home/pi/keyfile.key --start='UVX' --day=29 --text='MEET ME AT MIDNIGHT'
HPZXUWHMICMHUFCXOTU
pi@raspberrypi:~/bgneal-enigma-f3af458a5d2d $ sudo python pyenigma.py --key-file
 /home/pi/keyfile.key --start='APJ' --day=29 --text='QMMZIY'
UVXUVX
pi@raspberrypi:~/bgneal-enigma-f3af458a5d2d $ sudo python pyenigma.py --key-file
 /home/pi/keyfile.key --start='UVX' --day=29 --text='HPZXUWHMICMHUFCXOTU'
```

3 Encrypt/decrypt messages

Once you either have your plain text message ready to send or have received an encrypted message, the steps are the same. Follow the steps outlined in *Enigma Encryption* (on the second page of this feature) to encode or decode messages. For extra security, do as the Germans did and use the "day key" to encode the key for each message separately.

Build your own drone

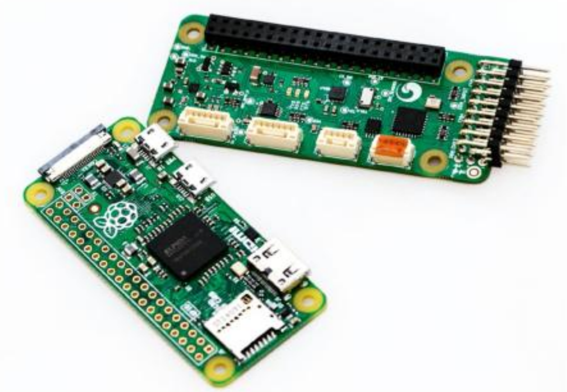
Linux is at the heart of commercial drones, which means that with the right kit and know-how you can build your own. **Alastair Jennings** explains how

What you need

- Raspberry Pi Zero
- Erle Robotics PXF Mini
- Erle Robotics PXF Mini Power Module
- HobbyKing Spec FPV250
- 100mm male to male servo cable
- FlySky-i6 controller
- Edimax AC EW-7811UAC
- RC XT-60 connectors

Quadcopters (drones) used to take hours of practice to master because even the simplest manoeuvres, such as takeoff and landing, could prove difficult. Learning to fly one took time and ultimately determination, and before you even took to the skies there was the small matter of constructing one.

Now that there's a good selection of pre-built and programmed drones on the market, you can go into major High Street retailers and buy one off the shelf. Drones such as the 3DR SOLO, Parrot Bebop and DJI Phantom have revolutionised the market, and slowly there are drones appearing with advanced flight features that make flying and controlling a drone much easier.



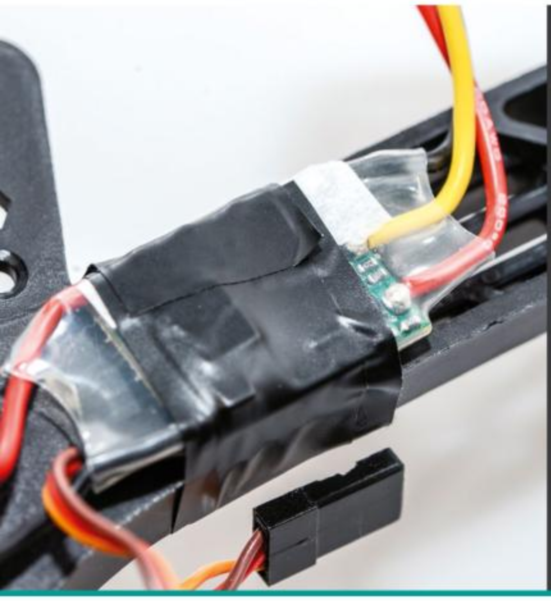
The big turning point in drone design was the addition of small processors on-board that were able to stabilise flight and implement advance features, such as auto-braking, takeoff and landing. These enabled the pilot to get on with having fun rather than worrying about the mechanics and programming.

As commercial drones have increased in popularity and become more widespread, so have the open hardware and software communities. The latest open source drones are challenging their more expensive rivals with advanced features, such as object avoidance and GPS navigation.

This challenge to the commercial models is no real surprise and you don't have to look too far into the DIY drone community to find out that many of the main manufacturers are extremely active in the open source world and regularly contribute advice and support to those wishing to build their »

WARNING!

This project is suitable only for those aged 18 or over. Piloting any drone is a dangerous and skilled task. Please seek suitable guidance and training before attempting to do so. Be aware there are serious legal consequences for failing to follow UK Civil Aviation Authority guidelines (see these at **www.caa.co.uk**).



GET THE AUTOPILOT LEVEL

Due to the small size of the drone it's quite difficult to keep things neat as you tuck in wires, wrap insulation tape and zip ties around the frame in order to hold everything together. Whilst it might seem fiddly and time-consuming to make sure that the PXFMini-Pi Zero combo is absolutely level, it's one of the most vital aspects of the build!

From the outset of building the frame you need to continually check that everything about your build is as accurate as possible.

The frame kit we've used in this project does the job well despite being cheap, but there are some of the parts, including the frame legs, that take a bit to fit correctly and have a habit of dropping out. The only way around this particular issue is to glue them in place – if one leg falls out during takeoff or landing, the effect as the drone tries to correct itself as can be quite dramatic.

Legs are an easy fix, but there are several other issues with the frame because it hasn't been designed for the shape or size of the autopilot we're using. Not only this but the size of the space for the autopilot is too small. This means that without filling the hole it's extremely difficult to get the autopilot to sit flat. If it's not flat then when it takes off it will try to level itself, and if you do get it into the air the flight will be erratic.

In order to get around this and also to help cut out vibrations from the motors, we added a large section of foam inside the slot. This helps to bulk out the section as well as create a rough damper for the autopilot to sit on.

own – for example, companies such as 3D Robotics sell autopilot systems that can be programmed through software applications such as Mission Planner.

3D Robotics' involvement in the community is apparent when you take a closer look at one of their drones. Exploring under the bonnet of the 3DR SOLO, you'll see that it's Linux based. The company also runs a huge education program, and a full SDK is available for the SOLO. We're going to take a look at building a basic drone from a recent open source community project, which you'll find at **http://bit.ly/LXFdrone**, with the latest Raspberry Pi Zero and Erle Robotics PXFMini.

SOLDERING IN THE PINS CAN BE A BIT OF A CHALLENGE BECAUSE OF THE SMALL SIZE OF THE BOARD

Setting up and soldering

The first step is to prepare the Pi Zero and PXFmini. Fitting the two together is relatively simple once the 40-pin GPIO connector has been soldered onto the Pi Zero. The connector is simply a set of two lines of pins that slot into the top of the Zero and the corresponding socket on the PXFMini.

The cost of the Pi Zero's basic board is low at just £4, but for this project you really need to get the starter kit that includes the unpopulated 40-pin GPIO connector, Mini USB and HDMI cables. A USB hub is also a good idea so that you can connect a keyboard, mouse and Wi-Fi dongle.

You'll actually end up with two 40-pin connectors if you buy this kit because, as it

› The motors might seem small but they're powerful. It's best to leave the propellers off until the last moment.

MULTISTAR
1704-1900

happens, there's one included with the PXFMini, but it's worth paying £8 for the other connectors along with the Pi Zero. We ordered our Pi Zero from **https://shop.pimoroni.com**.

Soldering in the pins can be a bit of a challenge due to the small size of the board, and you must also make sure that the pins have enough length above the board to interface with the PXFMini.

The easiest way to do this is to use some Blu-Tack (or a similar removable adhesive product). We also found that sanding the pins with a bit of Wet and Dry or fine emery cloth, as you prefer, just helps the solder to stick.

To prepare the board, place the GPIO pins into the Pi Zero and turn the board over so that you can see the pins coming through the board. Put it on the table so that it's at about a 45 degree angle and use a bit of Blu-Tack to secure.

Now push the pins so that there's only a small portion – a maximum of 1mm – appearing through the board, and use Blu-Tack to secure the position of the pins at one end of the board. The Blu-Tack needs to be positioned on pins that we're not soldering at first; then once one end of the pins is soldered and secured we can remove the Blu-Tack and finish the job. If the Blu-Tack gets hot then it will burn and become difficult to remove. You'll need to solder all the pins and make sure that you avoid any dry solders. A fine-tipped soldering iron with a new tip will make your life easier if you're not used to soldering.

Drone building

Once you have the two boards connected, you can make a start on the construction of the drone itself. There are many basic kits available, and they're straightforward to use. We've gone for a HobbyKing Spec FPV250.

Most of the small kits include all the basics, with a simple bolted-together frame, four small electronic speed controllers (ESC), and brushless motors along with a battery.

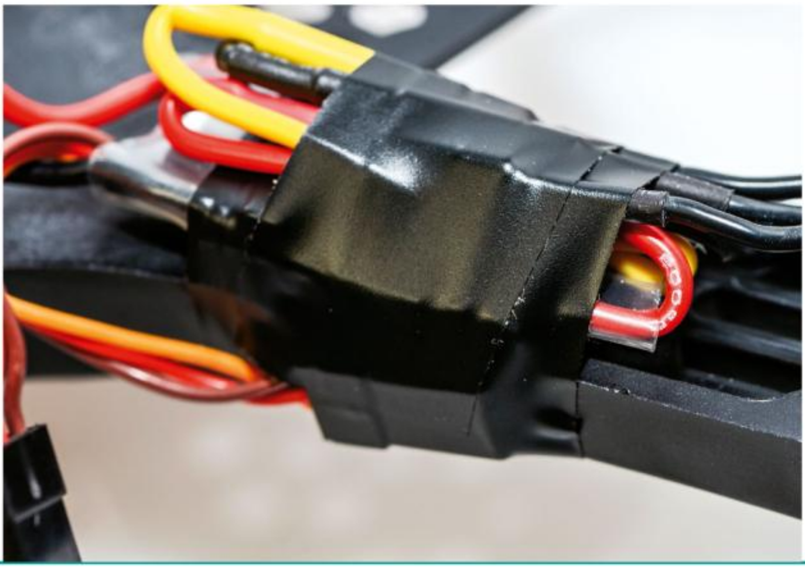
The main components and electronics all need to be taped and zip-tied to the frame, with only the motors requiring screws. At this point, it's well worth leaving the propellers off until the drone is correctly configured.

As you put the drone together there are a couple of key points. The PXFMini should be mounted with its connectors facing towards the front of the craft; these pins are used to connect the ESC and the receiver if you're using an RC unit.

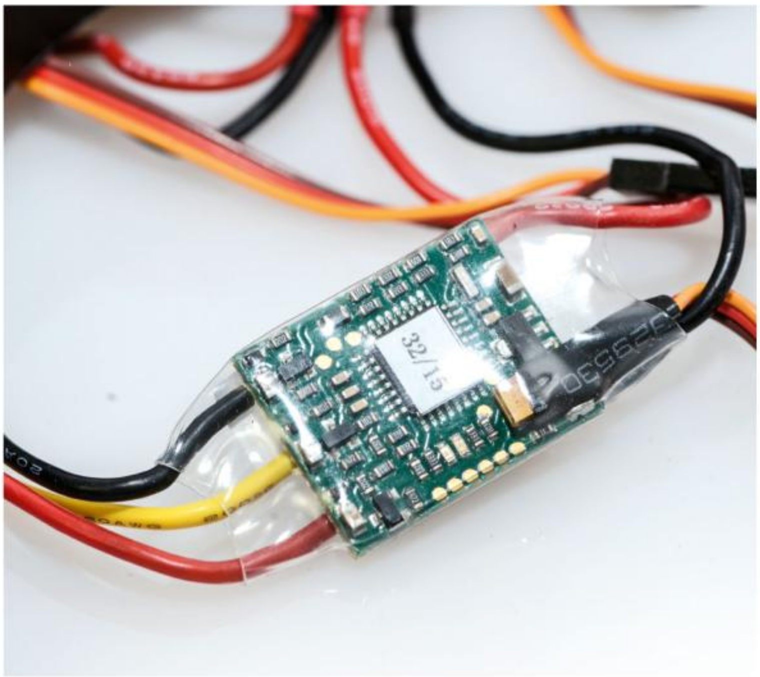
Between the Pi Zero and frame, it's essential that you add a section of foam to provide a bit of insulation from the vibration of the motors and to ensure that the PXFMini is level. A few additional extras that you'll need during this part of the build are one male-to-male servo extension and four

»

Each motor is controlled by a separate electronic speed controller (ESC).



Check that all wires are secure and that all motors react to commands from the transmitter before attaching the propellers.



Four speed controllers adjust the lift and direction of the drone.

male and female RC XT60 LiPo connectors. At the moment the two boards are empty of commands and before they can be used to control the drone they need to be flashed with an OS. This OS, as with all Raspberry Pi boards, is held on an SD card – a MicroSD card in this case – which can be quickly installed into the back of the Pi Zero.

Flashing the Pi Zero

The OS needed for the drone and for use with the Pi Zero and PXFMini autopilot system has been pre-compiled by Erle Robotics and is based on Debian. As long as you purchased the board directly from them, the company will email through a link to the latest version of the OS. If not, it's possible to compile it yourself, although this will be a more advanced task and might prove challenging if you're new to such things.

The Pi Zero can only read the FAT32 file system, so before you continue make sure you have formatted your card correctly and not used exFAT. With the OS downloaded and the microSD card ready, open a Terminal and type `df -h` to check the disks attached to your computer.

In order to flash the MicroSD card with the latest OS Image, you'll need to unmount the card. If your card is 32GB or more in capacity, you'll see that the disk has two partitions, so make sure you eject both. Type in:

```
umount /dev/disk2s1
```

Now to flash the OS to the card, make sure that the file you have downloaded has not yet been uncompressed and type in the following command (the initial `zcat` command first decompresses the disk image; change **/Path/to/image/** to your desired destination directory):

```
sudo zcat /Path/to/image/PXFmini.img.gz | sudo dd of=/dev/disk bs=8M
```

The flashing process can take some time as the uncompressed file is well over 7GB.

Once the process is complete, eject the card from your computer and install it into the Pi Zero. The Pi must now be connected to a monitor, keyboard and mouse to finish the installation process. After the Pi boots, you'll see the basic Erle Robotics splash screen and a bar of Icons showing the different Vehicle projects that the board can be used with.

Click the Erle Copter picture and the screen will disappear and the board will reboot. Leave the Pi Zero to run through the boot sequence and this time rather than loading a graphical user interface, the board will boot to a command line

THE FIRST FLIGHT

Before you begin, double-check your drone and make sure that everything is in place and all wires are secure and taped to the frame. Because this will be your first attempt at flying the drone, remove the propellers so that if anything does go wrong the drone will at least stay exactly where it is rather than bouncing around the room poking eyes out and smashing ornaments.

Start by switching on the controller, making sure that all switches are in the up position and the throttle is in the lower position. Now connect the battery to the power module and you should hear a beep as the board starts loading. This can take up to a minute to complete. Another beep will signal the end of the boot sequence, and then another 5 to 10 seconds is needed before the drone is ready for testing.

On the transmitter, move the left stick to the bottom-right position for five seconds and, holding it in place, flick down switch D (SWD), which on the SKYFLY FS-i6 is the switch on the top right of the handset. The motors should now start. Flick the switch back up to stop the motors. Now you should be able to use the throttle to start and increase the speed of the motors. If you decide to leave the drone for a short time then the start-up process will need to be repeated.

If all is in order and all four motors spin and react correctly with the transmitter, then you're ready to start your first flight once the propellers are attached. If only three motors start or you can visibly see that one motor is spinning at a much lower speed than the others, then you'll need to load AMP and configure your transmitter with the drone first.

showing that the installation has completed and the board can be unplugged.

At this point the drone should be ready for its first flight test. To get it to fly you'll need to connect it to some type of control device such as a radio control, Bluetooth or Wi-Fi with ROS (Robot Operating System). You do need to take into consideration that this is an open hardware project and the components that we'll be using will be different from those used by Erle Robotics. For complete ease, we've opted for the traditional radio-controlled option and used a FLYSKY FS-i6 with the new FS-iA6B receiver, which we located at Maplin for £50. The important factor here is that the receiver is of the PPM type mentioned earlier.

PPM enables several servos – or in this case several ESC – to be connected to one port and controlled individually. This works in just the same way as traditional servos would, with an individual port in the receiver for each. This cuts down on the amount of wires needed but more importantly it's the hardware required by the PXFMini in order to interface with the controller. Once you've connected everything it should work as described here, although you may find that some fine-tuning of the transmitter controls is needed.

The Pi Zero doesn't feature the ability to network over the USB ports, so a Wi-Fi dongle is required in order to connect to your machine. We've used a dongle bought directly from Erle Robotics at 55 euros. You'll need to make sure that your machine has a 5GHz compatible Wi-Fi device in order to get things to work properly. Once connected you can download and install AMP Planner from **http://ardupilot.org** and run through the calibration and setup process in order to get your drone to work correctly. ■

If there's one thing more satisfying than flying a drone, it's flying a drone you built yourself.

ONCE YOU CONNECT EVERYTHING IT SHOULD ALL WORK, WITH SOME FINE-TUNING OF THE CONTROLS

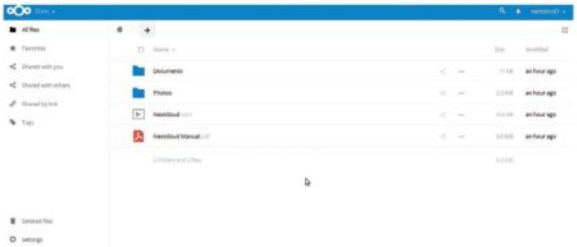
Set up a Raspberry Pi as a private server

Nate Drake shows you how to set up your own free Dropbox-style cloud service on the Raspberry Pi to securely store and share your data

Dropbox, Google Drive and iCloud are just a few of the myriad of online data storage and file transfer services available. The days are long gone when you stored your data on one computer and hoped for the best. Even backing up your precious files to an external hard drive, which at least protects to some extent against the risk of data corruption or hardware failure affecting your computer's hard disk, doesn't cut it any more. Now you can back up your data and share files with others via "the cloud".

The terminology is romantic but imprecise. All it means is that the servers on which you store your stuff are accessed over an internet connection. The dozens of "cloud storage" services out there each promise a certain amount of free space on their servers in which to store your personal data.

One of the most recent offerings is from Nextcloud, an open source fork of the ownCloud project which was started in April 2016. Nextcloud, like its predecessor ownCloud, strongly matches the functions of services like Dropbox and Google Drive in allowing you to sync data, collaborate on documents and share files.



› Anyone familiar with the layout of services like Dropbox or Google Drive will be able to use Nextcloud right away.

However, Nextcloud's software can be installed on a private server, leaving you in control of your data, without having to pay monthly or annually for storage space. Apps can be used to expand Nextcloud's functions to include calendars, editing documents and much more.

The Raspberry Pi is a perfect candidate for Nextcloud's software because it uses very little power when run around the clock like a web server. For this project we will go through the preparations you'll need to ready the Pi for Nextcloud, then the steps to install the software itself.

Nextcloud preparations

Although installing the Nextcloud software itself is quite easy, you will need to prepare your Pi.

In the first instance, it's best to have a dedicated Raspberry Pi to use with Nextcloud and for no other purpose. Considering that you're relying on it to keep your data safe, it's unwise to risk any other app interfering with the process.

You'll next need to set up a static IP address for your Pi so that other devices can find it. The steps outlined in the tutorial are for a Pi that's connected to your router via Ethernet cable. This is optional but will result in a faster and more secure

SECURE YOUR NEXTCLOUD

If you plan to access Nextcloud over the web, consider requesting a legitimate certificate from a certificate authority. You can obtain an SSL certificate free of charge from websites such as Let's Encrypt (**https://letsencrypt.org**) or Start SSL (**https://startssl.com**). Otherwise you can generate a self-signed certificate (which, naturally, will be used only for accessing Nextcloud...).

First you need to tell Apache to use it and where to find it. Begin by opening Terminal on your Pi or connecting via SSH and running the command `sudo su -` to elevate your user privileges. Next run each of these commands, pressing Return after each to prepare Apache :

```
a2enmod ssl
a2enmod rewrite
```

Next create a directory for your certificates:

```
mkdir -p /etc/apache2/ssl
```

Now generate the SSL certificate yourself if necessary:

```
openssl req -new -x509 -days 3650 -nodes -out /etc/apache2/ssl/nextcloud.pem -keyout /etc/apache2/ssl/nextcloud.key
```

You'll be asked to answer a series of questions about your country, email address, etc. Press Return for each question to give the default answer for each except "common name". Here you should type the IP address of your Pi – in our case, **192.168.1.118**.

Next create an empty configuration file for Nextcloud for Apache to use:

```
sudo nano /etc/apache2/conf-available/nextcloud-ssl.conf
```

Paste in the following:

```
<VirtualHost *:80>
  RewriteEngine on
  ReWriteCond %{SERVER_PORT} !^443$
  RewriteRule ^/(.*) https://%{HTTP_HOST}/$1 [NC,R,L]
</VirtualHost>
<VirtualHost *:443>
  ServerName 192.168.1.118
  SSLEngine on
  SSLCertificateFile /etc/apache2/ssl/nextcloud.pem
  SSLCertificateKeyFile /etc/apache2/ssl/nextcloud.key
  DocumentRoot /var/www/html/

  <IfModule mod_headers.c>
    Header always set Strict-Transport-Security “max-age=15768000; includeSubDomains; preload”
  </IfModule>
 </VirtualHost>

```

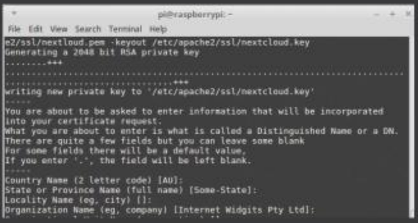
Press Ctrl + X, then Y, then Return to save and exit.

Enable the Configuration file with:

```
sudo a2enconf nextcloud-ssl.conf
```

Finally, restart Apache with the following command and you're good to go:

```
sudo service apache2 restart
```



```
e2/ssl/nextcloud.pem -keyout /etc/apache2/ssl/nextcloud.key
Generating a 2048 bit RSA private key
........+++
.............................................................................
.................................+++
writing new private key to '/etc/apache2/ssl/nextcloud.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
```

Make sure to set the "Common Name" to the IP address of your Pi.

connection. If you want to use wireless, simply substitute "wlan0" in place of "eth0" when modifying your network configuration in Step 1 of our guide on p68.

Next you'll need to decide on your database type. If you plan to be the only person using Nextcloud, have only a few files and want a quick and easy setup, simply follow the steps outlined in our guide to use Nextcloud's built in database software, which employs SQLite. If you want to make sure Nextcloud can handle larger files and/or multiple users, consider using a MySQL database by following the steps in *Using MySQL with Nextcloud* over the page.

The next steps involve installing software on your Pi to turn into a web server. In this case we are using Apache, which is the most popular web server software on the internet.

Once Nextcloud has been installed via the command line, you'll be able to navigate to it in your web browser to complete the setup process using the handy Installation Wizard.

Nextcloud clients

The first message you'll see after creating your account on Nextcloud is a link to download a Nextcloud client for your mobile device or computer. This is simply so that you can sync your files between your devices and the Pi.

At the time of writing clients are available free of charge for Windows, Mac and Linux. There's also a free Android client in the Google Play Store. The iTunes App Store also has a Nextcloud client for iPhones, iPods and iPads priced around 85 pence.

Once Nextcloud is up and running and you navigate to your Admin page, you may see several messages. None of them will stop you from using Nextcloud right away to upload files and enable Apps (see over the page for more about these) but it's important to understand them. Some of them alert you to important security considerations, which you should give some serious thought to and if necessary act upon before you begin using Nextcloud and expose your personal data to risks that are entirely avoidable.

Post-install tweaks

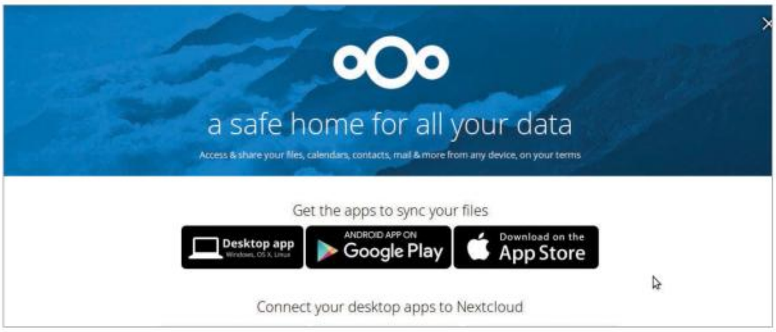
The first message in red reads "Your data directory and your files are probably accessible from the Internet." You can test whether or not this is true by visiting **http://192.168.1.118/nextcloud/data/yourusername/files/** to see a list of your files.

Naturally if you want to be able to access your Nextcloud from outside your home network, this may be the way you want things. One way to make sure that your data is safe, though, is to enable server-side encryption, which will scramble data inside your files, making them useless to anyone who downloads them without your password.

Click on Files at the top left of the screen, then Apps, then Not Enabled. Scroll down to the Encryption app and click Enable. Next click your username at the top right of the screen and select Admin. Click the Encryption tab which has now appeared in the left-hand menu. Tick the box marked Enable Server Side Encryption and read the wordy disclaimer in red. Once you feel you've been duly warned, click on Enable Encryption. »

Pi bites

Be aware that encryption is not retrospective – that is, it will encrypt only new files you place in Nextcloud after it's enabled. Log in and log out again after enabling encryption to get started.


a safe home for all your data

Access & share your files, calendars, contacts, mail & more from any device, on your terms

Get the apps to sync your files

Connect your desktop apps to Nextcloud

The first time your run Nextcloud you'll see links to the various client apps for your devices. They can also be downloaded from https://nextcloud.com/install/.

If you don't wish to encrypt your files or you just want to share them over your home network, you can move your Nextcloud data directory to a safe location where the Apache Web Server cannot access it.

If you decide to do this, first open Terminal on your Pi or connect via SSH and stop Apache altogether with the command

```
sudo systemctl stop apache2
```

Next decide on a folder where you want to keep your Nextcloud data – for example **/home/pi/nextcloud-data**. Run the following command to move the data directory to the new location:

```
sudo mv /var/www/html/nextcloud/data /home/pi/
nextcloud-data
```

Now you'll need to tell Nextcloud where to find the new data directory. Run this command to open Nextcloud's configuration file:

```
sudo nano /var/www/html/nextcloud/config/config.
php
```

Using the arrow keys scroll down until you find the text **'datadirectory' => '/var/www/html/nextcloud/data'**. Amend it to the new location, for example something like the following:

'datadirectory' => '/home/pi/nextcloud-data'

Press Ctrl +X, then Y, then Return to save and exit.

Finally make sure that Nextcloud can still access the Data Directory with this command:

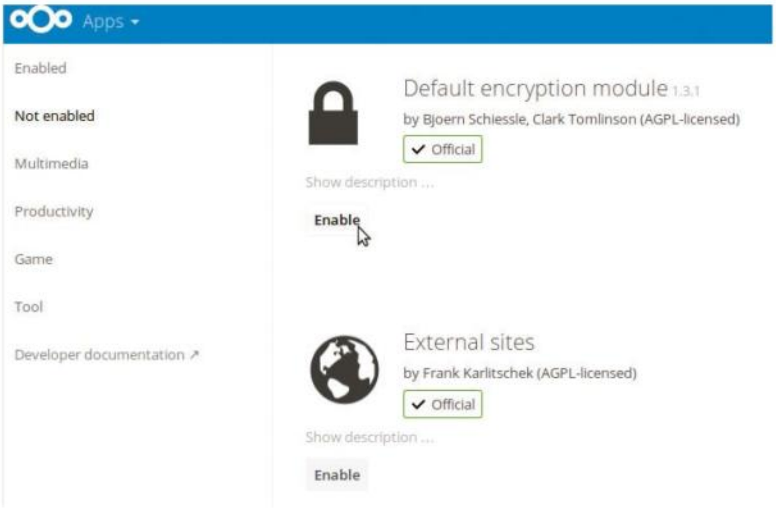
```
sudo chown www-data:www-data /home/pi/
nextcloud-data
```

Restart the Pi for your changes to take effect.

Incidentally, this method is also an excellent way to back up your Nextcloud Configuration and files or move them to a new computer.

If you revisit the Admin Panel, you might also be told that you're accessing it over HTTP rather than HTTPS. If you plan to access Nextcloud only over your home network, this isn't very worrisome. However, if you want to make sure that your connections between your devices and Nextcloud are encrypted, be sure to follow the steps in *Secure your Nextcloud* on the previous page.

Enable the Encryption Module to encrypt each file using ultra secure 256-bit AES in future. Be sure to read the disclaimer before continuing.



Default encryption module 1.3.1

by Bjoern Schiessle, Clark Tomlinson (AGPL-licensed)

External sites

by Frank Karlitschek (AGPL-licensed)

When securing Nextcloud you'll need SSL certificates which encrypt your connection. These are usually obtained from a certificate authority but you can generate one yourself on the Raspberry Pi if you wish. These "self-signed" certificates are not generally considered to be as secure compared to those offered by a third party because anyone can generate one. However, since you are the one creating the certificate, you know it's trustworthy.

If you use the Nextcloud App on your mobile device or access it via your browser, you may see a security warning saying that the connection isn't trusted. Rest assured this is perfectly normal. You can add a Security Exception for your Nextcloud instance and the client or browser will remember it going forward.

The final error message you might see is that "No memory cache has been configured." In theory the memory cache contains versions of recent files to allow Nextcloud to access your data more quickly. In practice we have not noticed a huge improvement in performance from configuring the Pi to work with a memory caching program. If you are interested in setting this up, you can find a list of caching programs and instructions at **https://docs.nextcloud.com/server/10/admin_manual/configuration_server/caching_configuration.html**. The website asks you to check which version of php you are running before choosing a program, which you can do using the following command:

```
php -v
```

Nextcloud Apps

By default you can upload and arrange files in Nextcloud as well as see your images laid out in a basic Gallery. This is perfectly adequate for day-to-day use but there are a number of official and unofficial apps which can be used to enhance your cloud. The Encryption app if you enabled it earlier is a great example.

You can access the list of available Apps by clicking on the text at the top left of the screen – Files then Apps, then Not Enabled. Click on "Show Description" beside an app to find out more.

In the illustration we see the External Storage Support app, which allows syncing with regular cloud providers such as Google Drive, Dropbox or even other Nextcloud/ownCloud servers.

Whichever app you choose, click Enable to start using it. The app will automatically download and install. Usually it can then be accessed by clicking on Apps at the top left of the screen or, as is the case with the External Storage app, by going to your Admin page and selecting it from the menu.

Clicking on the settings icon in the Apps page will allow you to enable "Experimental Apps". As the name suggests, these are apps which either are still in development or have been created by third parties and not fully tested with Nextcloud.

If you choose to enable them, refresh the page and scroll down past the official apps to see what's on offer. Bear in mind that the experimental apps may interfere with the running of others – for example the enhanced Gallery app Gallery+ will require you to disable the default app.

Storm Clouds

Provided that the required files are downloaded in advance, the setup process for Nextcloud is quite simple. If things do go wrong post-install, it's best to start the process over from scratch by removing the Nextcloud configuration file with the following command:

```
sudo shred -u /var/www/html/nextcloud/config/config.php
```

When you next navigate to your Nextcloud instance in your browser – for example **http://192.168.1.118/nextcloud** – you will see the setup screen again and can create a new admin account. If you have encrypted your documents previously, you'll need to use the same password as you did with your previous account.

If you choose to connect to Nextcloud over HTTPS, you might see an error message after enabling it about "Strict Transport Security". This is an optional setting which means that no device can connect to your Nextcloud in a non-secure manner. However if you followed the steps outlined in *Secure your Nextcloud* (previous page), anyone attempting to connect via regular HTTP will be forced to use the more secure HTTPS anyway, so this message can be ignored. If it's simply annoying you, follow the steps outlined in the "Hardening and Security Guidance" guide in the section marked "Enable HTTP Strict Transport Security". See **https://docs.nextcloud.com/server/10/admin_manual/configuration_server/harden_server.html**.

If you have a large number of files then the available space on your SD card may fill up quickly. If you need more space urgently, the simplest solution is to copy your Nextcloud data folder to another SD card with more storage space. Technically you can keep your data folder on an external hard drive or USB stick but this can cause permissions issues with Nextcloud, so do this only if you are confident both with formatting drives and modifying file permissions.

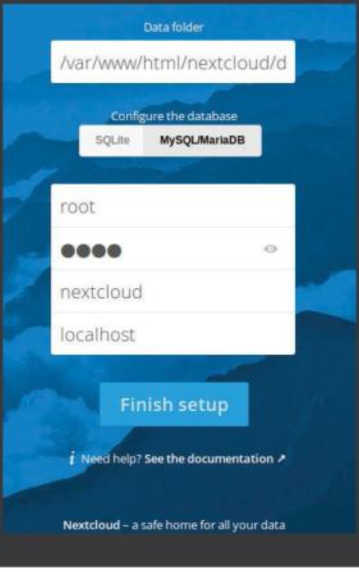
If you want your Nextcloud instance to be accessible over the internet, you will need to obtain a static IP for your router. You can either pay a fee to your ISP if they offer this service or use a service like **www.noip.com** to get a permanent website address such as **http://davesnextcloud.ddns.net**. Next you will need to forward port 80 (or port 443 if you're using SSL) on your router to your Pi's IP address. The steps for each router are different, so we cannot provide instructions here, but the website **www.portforward.com** has some useful information for common routers.

USING MYSQL WITH NEXTCLOUD

By default Nextcloud uses SQLite to create and store your files in a database. SQLite is quick and easy to set up and, if you plan to be the only person using Nextcloud and don't plan on uploading a large number of files, then you're welcome to skip this step. MySQL is a little more difficult to set up but it is much more efficient than SQLite and can handle many people using Nextcloud at the same time.

To set up, open Terminal on your Pi or connect via SSH and install all the necessary prerequisites with the following command:

```
sudo apt-get install php5-fpm php5-mysql openssl ssl-cert php5-cli php5-common php5-cgi php-pear php-apc curl libapr1 libtool php5-curl libcurl4-openssl-dev php-xml-parser php5-dev php5-gd libmemcached* memcached php5-memcached -y
```


Data folder
Configure the database
SQLite
MySQL/MariaDB
Finish setup
Need help? See the documentation ↗
Nextcloud – a safe home for all your data

/var/www/html/nextcloud/d

root

●●●●

nextcloud

localhost

Next install the MySQL software itself by entering the following:

```
sudo apt-get install mysql-server -y
```

You'll be asked to type a password for MySQL root user – for example use the word "root".

Now restart the Pi and open MySQL with the command:

```
mysql -u root -p
```

Enter the password you just created. Next run the following commands to create a database named "nextcloud". Press Return after each command:

```
CREATE DATABASE nextcloud;
CREATE USER nextcloud@raspberrypi IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON nextcloud.* TO nextcloud@raspberrypi;
FLUSH PRIVILEGES;
exit
```

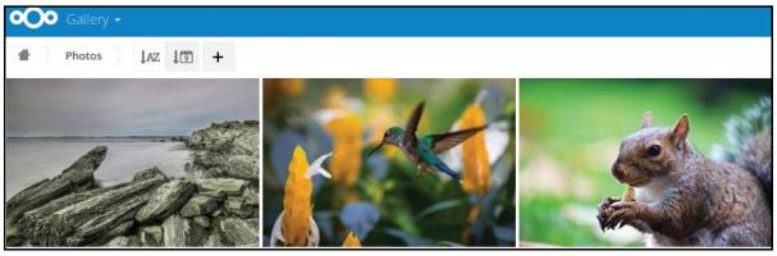
Feel free to change the password from "password" if you wish – it certainly doesn't represent the level of security we would recommend.

When you get to Step 6 of the setup process in the guide on the next page, before creating an Admin account click Storage and Database, then MySQL/MariaDB. Enter the username and password "root" (or whichever password you chose when setting up MySQL). The database name is "Nextcloud".

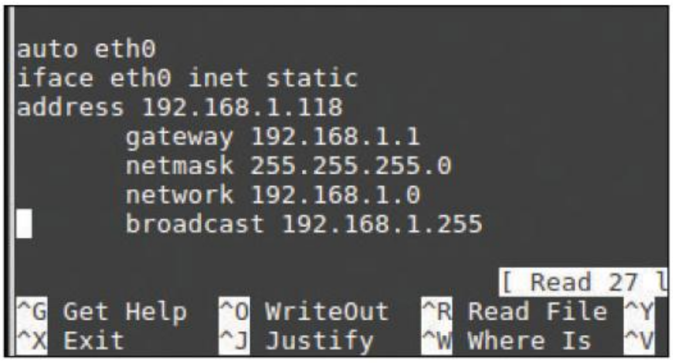
Scroll down on the setup screen before creating your admin account and click on MySQL/MariaDB. Enter the root username and password, as well as the name of your database.

A final word of caution is to maintain regular backups of your Nextcloud data. When you're using large services like Dropbox or Google Drive, several copies of your data are stored in various servers around the world. You most likely will only use one Raspberry Pi, so keep it somewhere safe. Consider using the External Storage app to place a copy of your files in a Dropbox account or, better yet, copy your "data" folder onto an external drive from time to time and store it offsite. »

The default Gallery app must be disabled for the experimental app Gallery+ to work properly. To disable an app, click on Enabled next to the app's name.



SET UP NEXTCLOUD ON YOUR PI



```
auto eth0
iface eth0 inet static
address 192.168.1.118
        gateway 192.168.1.1
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255

                                        [ Read 27 l
^G Get Help   ^O WriteOut   ^R Read File  ^Y
^X Exit       ^J Justify    ^W Where Is   ^V
```

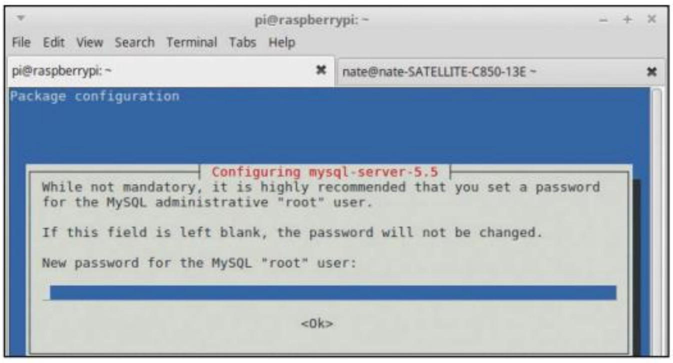
1 Assign Static IP

Connect to the Pi over SSH or open Terminal and run the command:

`sudo nano /etc/network/interfaces`

Find the line "iface eth0 inet manual". Delete it and replace with the seven lines shown in the screenshot above. We are assuming here that the Pi is connected over Ethernet. If you want to use wireless, simply substitute "wlan0" instead of "eth0". To apply the changes, run the command:

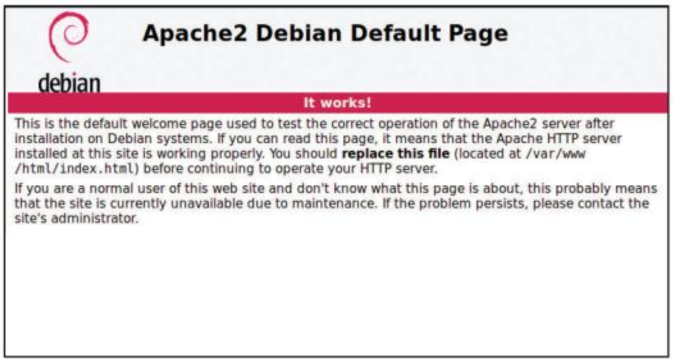
`sudo /etc/init.d/networking restart`


pi@raspberrypi: ~
File Edit View Search Terminal Tabs Help
pi@raspberrypi: ~
nate@nate-SATELLITE-C850-13E ~

```
Package configuration

         ┤ Configuring mysql-server-5.5 ├
 While not mandatory, it is highly recommended that you set a password
 for the MySQL administrative "root" user.

 If this field is left blank, the password will not be changed.

 New password for the MySQL "root" user:

                              <Ok>
```

2 Install MySQL server

This step is optional but recommended if you plan to have multiple users and/or you want to store a large number of files in Nextcloud. MySQL is better suited for this than SQLite, which is what Nextcloud uses by default to create and store files in a database. To install MySQL, follow the instructions in *Using MySQL with Nextcloud* on the previous page.



Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

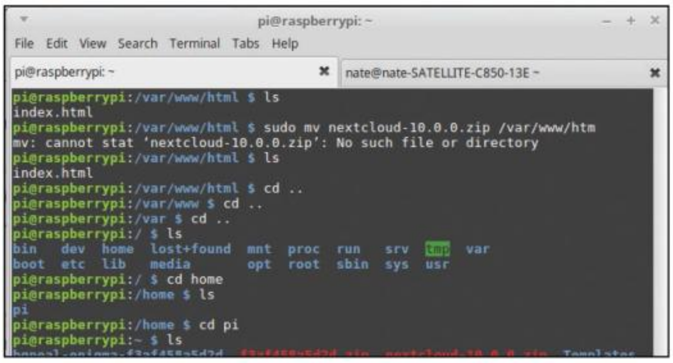
3 Install required software

To install Apache, PHP and other required software, run the command:

`sudo apt-get install apache2 php5 php5-gd sqlite php5-sqlite php5-curl openssl`

Then restart Apache with `sudo service apache2 restart`

You can check that Apache is working properly by opening a browser and going to **http://youripaddress** – for example, **http://192.168.1.118**. If all is working you'll see a splash page saying "It works!"


pi@raspberrypi: ~
File Edit View Search Terminal Tabs Help
pi@raspberrypi: ~
nate@nate-SATELLITE-C850-13E ~

```
pi@raspberrypi:/var/www/html $ ls
index.html
pi@raspberrypi:/var/www/html $ sudo mv nextcloud-10.0.0.zip /var/www/htm
mv: cannot stat ‘nextcloud-10.0.0.zip’: No such file or directory
pi@raspberrypi:/var/www/html $ ls
index.html
pi@raspberrypi:/var/www/html $ cd ..
pi@raspberrypi:/var/www $ cd ..
pi@raspberrypi:/var $ cd ..
pi@raspberrypi:/ $ ls
bin   dev  home  lost+found  mnt  proc  run   srv  tmp  var
boot  etc  lib   media       opt  root  sbin  sys  usr
pi@raspberrypi:/ $ cd home
pi@raspberrypi:/home $ ls
pi
pi@raspberrypi:/home $ cd pi
pi@raspberrypi:~ $ ls
```

4 Download Nextcloud software

To download the Nextcloud Zip file, run the following:

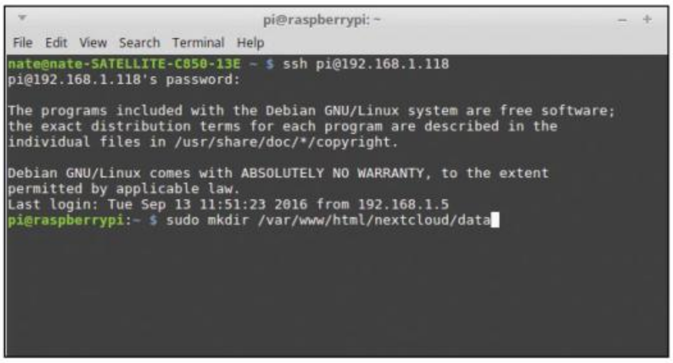
`wget https://download.nextcloud.com/server/releases/nextcloud-10.0.0.zip`

Move the file where it needs to go, then move to the directory yourself:

`sudo mv nextcloud-10.0.0.zip /var/www/html`

`cd /var/www/html`

To extract the files, use `sudo unzip -q nextcloud-10.0.0.zip`



```
nate@nate-SATELLITE-C850-13E ~ $ ssh pi@192.168.1.118
pi@192.168.1.118's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Sep 13 11:51:23 2016 from 192.168.1.5
pi@raspberrypi:~ $ sudo mkdir /var/www/html/nextcloud/data
```

5 Create Nextcloud data directory

If you want your files to be accessible over the web, place your data directory inside Nextcloud's folder using the following:

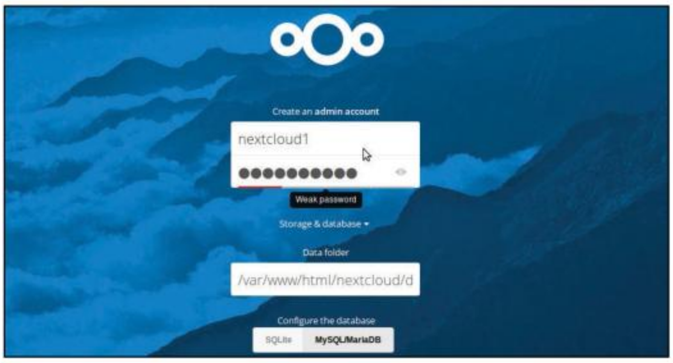
`sudo mkdir /var/www/html/nextcloud/data`

Otherwise use a location on your Pi, for example:

`sudo mkdir /home/pi/nextcloud-data`

To make sure there are no permissions issues, run the following:

`sudo chown -R www-data:www-data /var/www/html/nextcloud`



Create an admin account

nextcloud1

●●●●●●●●●●

Weak password

Data folder

/var/www/html/nextcloud/d

6 Launch setup from a browser

Do the same for your data directory – for example:

`sudo chown www-data:www-data /var/www/html/nextcloud/data`

Now to get going, open a web browser connected to your network and visit **http://192.168.1.118/nextcloud** to visit the setup screen. You'll be asked to create an Administrator Username and password. Before you do, check that the path to the data directory is correct. You also may want to click on "Storage and Database" to use MySQL. ■

Build your first Pi-powered robot

Les Pounder runs through how to build a simple but elegant, budget-busting custom robot for scaring the cat. Just because

Robotics is an exciting way to introduce people to programming but it can also be a little difficult sometimes for newcomers to get to grips with, as well as being expensive. Enabling someone to create an easy-to-build and cost-effective robot is a significant step in their learning. So in this project we'll build our own robot and create a Python 3 library that enables anyone to control it. For this project you will need: any model of Raspberry Pi; Raspbian (**www.raspberrypi.org/downloads**), a Wi-Fi dongle and Pi connected to your home router; a USB battery pack, a robot chassis kit (**http://bit.ly/LXF203_Robot_kit**), an L298N motor controller (**http://bit.ly/LXF203_L298N**); four AA batteries and some Blu-tack.

Building a robot chassis is a great activity and the kit (mentioned above) comes with everything you need to get started. You will need to solder the red and black wires to the motor terminals – if you can't solder then now is a great time to learn from a friend or a local hackspace.

With the chassis built, we now focus on the motor controller, which is an L298N H bridge controller. An H bridge enables a motor to go forwards and backwards. Our L298N has two outputs for our motors: the left side is served by OUT1 and 2, the right by OUT3 and 4. Connect the wires from your motors to these terminals and ensure they are secure. Our AA battery pack connects to +12V and GND terminal. We also need to connect one of the GND from our

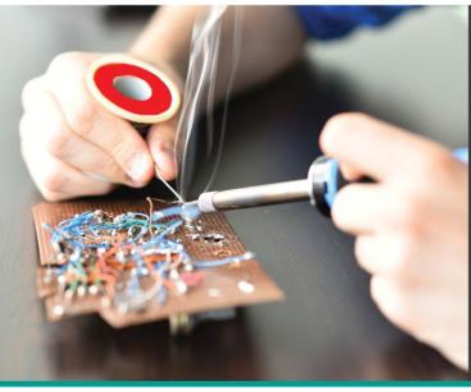
»

REMOTE CONNECTION

We set up an SSH server on our Raspberry Pi at the start of this project, so now let's use it to remotely control our robot. In Linux we can run the `ssh` command from the terminal. To SSH into our robot we need to know its IP address, which we wrote down earlier, and the name of the user – typically **pi** for a Raspberry Pi. We then type the following into a terminal to proceed:

```
$ ssh pi@IP ADDRESS
```

Replace `IP ADDRESS` with your Pi's IP. You'll be prompted for the Pi password, which is typically **raspberry**, and once logged in, any command you enter will output on your Pi. Navigate to the directory where you saved the **robot.py** and **test.py** files. To run the test code in the terminal type `sudo python3 test.py`. Press Return and the robot will come to life and perform the test sequence. It works! But let's open an interactive Python 3 session and live code the robot with `$ sudo python3 -i`. We can now import the robot library and run the same functions as per the **test.py** file. To return to the terminal just press Ctrl+d.



SOLDERING

For this project we bought a robot chassis kit from eBay that included two DC motors. These motors come assembled but require soldering two wires to the terminals for power. Soldering is an essential maker skill and it is really easy to learn, though adult supervision is essential for our younger would-be solders out there. There are many helpful YouTube tutorial videos, but the best we've found is from Carrie Anne Philbin (**http://bit.ly/LXF203_Solder**).

Cheap soldering iron sets can be bought for around £10, but it's worth investing in something better than a very basic, bargain-basement model. The Antex XS25, available for around £25, is a great starter to intermediate soldering iron. Soldering should be undertaken in a spacious, well-ventilated room with a clear workspace. Soldering is great fun and your local hackspace/LUG can help you to learn in a safe manner.

Pi bites

Securing your components to the chassis is important, otherwise your Pi will be dragged along by the robot. We used Blu-tack but for a long-term project cable ties are better.

Raspberry Pi to the L298N GND terminal. On the L298N we can see four pins marked IN1 to IN4. These are inputs that we use to connect the L298N to our Raspberry Pi's GPIO (General Purpose Input/Output) pins.

By turning a GPIO pin on or off we can trigger the input pins accordingly and control the motor direction. We connected our inputs to the following GPIO pins: IN1 to 17, IN2 to 22, IN3 to 18 and IN4 to 23. We used the Broadcom pin mapping, a standard set by the Raspberry Pi Foundation. A great reference for the GPIO is **http://pi.gadgetoid.com/pinout**, which explains all you'll need to know about Broadcom pin mapping.

Software setup

Boot your Raspberry Pi to the desktop and open a terminal – you can find the icon in the menu bar at the top left corner of the screen. In the LXTerminal type the following and press Return to run:

```
$ sudo raspi-config
```

Using the arrow keys, navigate to Advanced Options and press Return. In the Advanced menu navigate to the SSH Server option, press Return and in the new screen choose to Enable the SSH server. Exit from the menus and reboot your Raspberry Pi. Reboot back to the desktop, open another LXTerminal and type the following to get your IP address and write the address down:

```
$ hostname -I .
```

In the same terminal type the following to launch the Python 3 editor with superuser powers:

```
$ sudo idle3 &
```

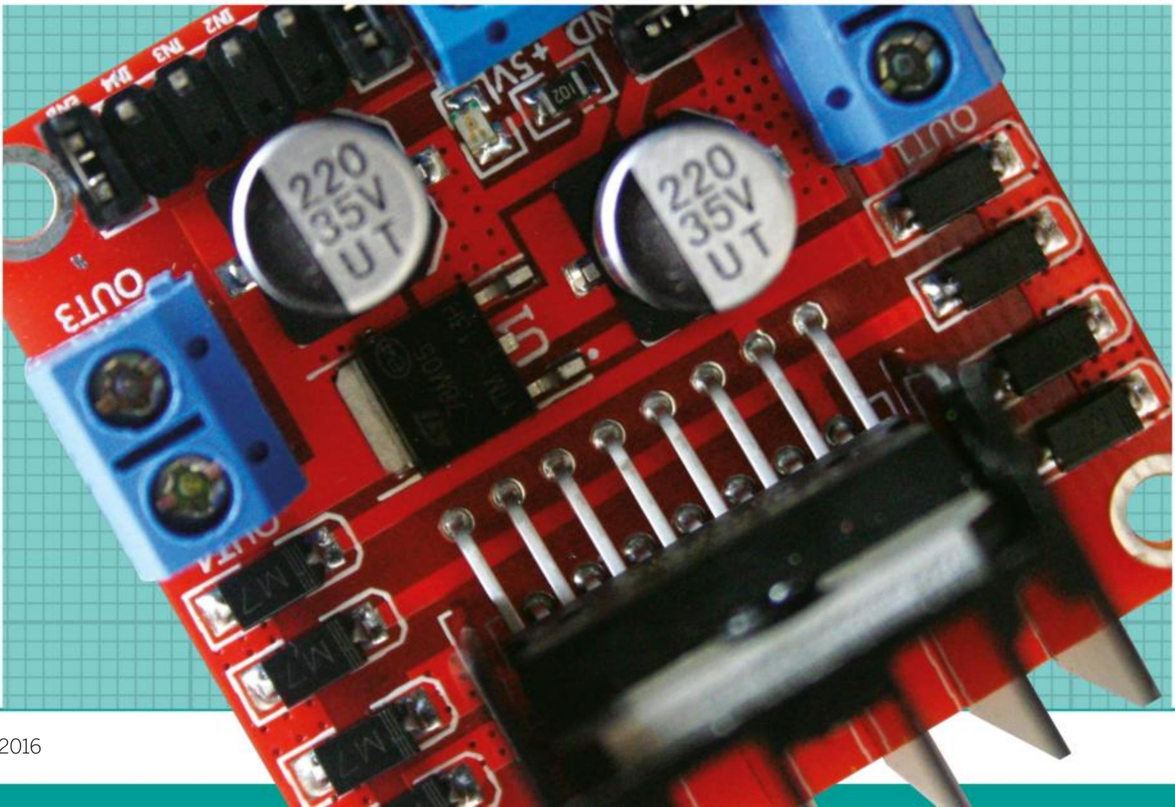
We'll start our code by importing two libraries. The first enables our code to talk to the GPIO pins on our Pi while the second provides the time library:

```
import RPi.GPIO as GPIO
import time
```

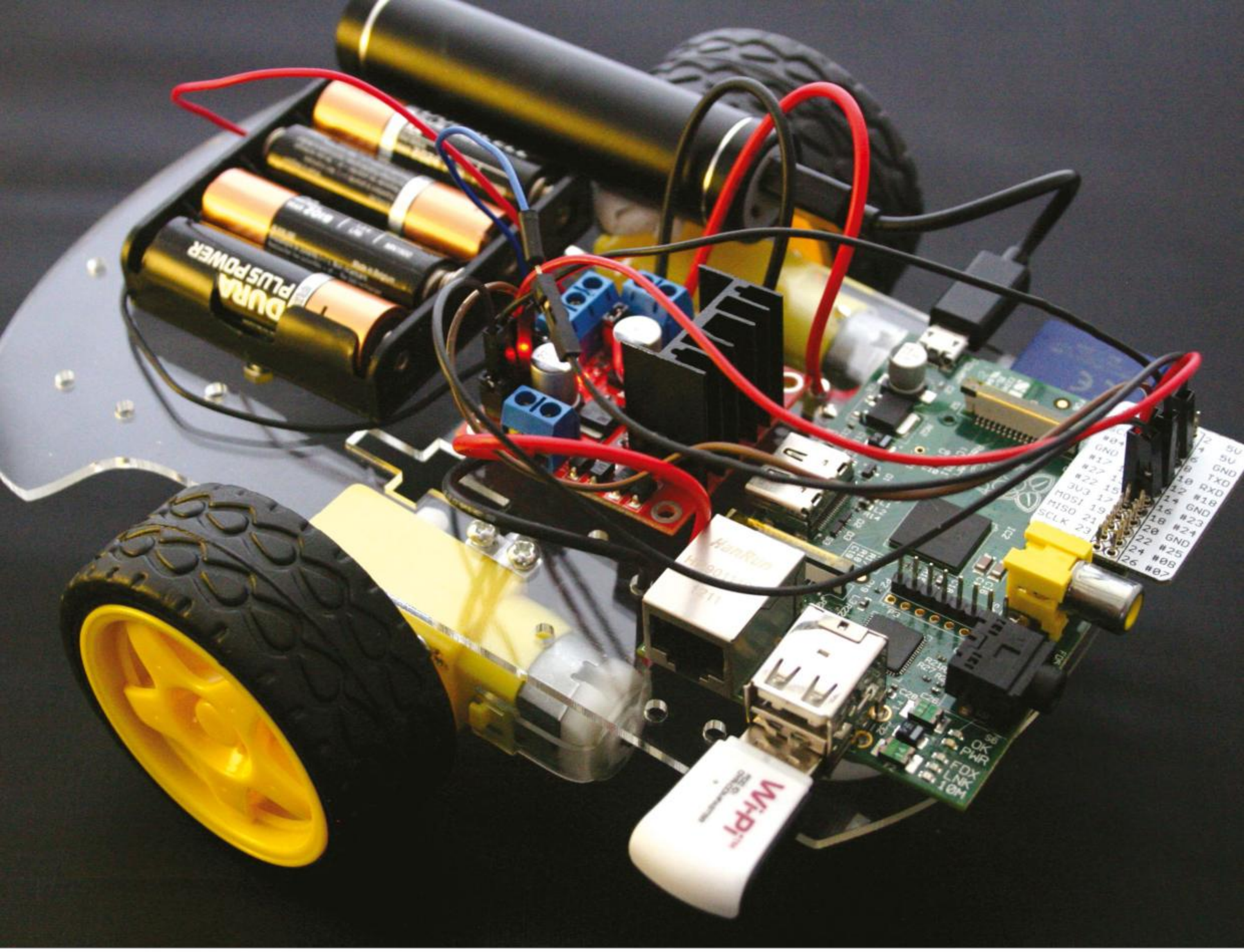
When using the GPIO pins we will refer to them using their Broadcom pin numbering and we must, in turn, configure our code to use those numbers with `GPIO.setmode(GPIO.BCM).` Rather than refer to each pin throughout our code we'll create four variables to store the GPIO pin connected to each of the inputs on the L298N:

```
fwdleft = 17
fwdright = 18
revleft = 22
revright = 23
```

In order to use each GPIO pin we need to instruct the code what each pin will be: an input or output. As we will be sending current from the GPIO pins they will be an output. So using a list, known in other languages as an array, and a **for**



› **The screw terminals enable connections between the batteries, Raspberry Pi and motors.**



Our finished robot is fun on two wheels!

loop, we shall iterate over each item in the list, which are our variables, and configure each GPIO pin as follows.

```
motors = [fwdleft,fwdright,revleft,revright]
for item in motors:
        GPIO.setup(item, GPIO.OUT)
```

Driving our robot

We now create four functions that will handle driving our motors in a particular direction. Each of the functions will take an argument, a duration of time that's expressed as an integer or a float:

```
def forward(i):
        GPIO.output(fwdright, True)
        GPIO.output(fwdleft, True)
        time.sleep(i)
        GPIO.output(fwdright, False)
        GPIO.output(fwdleft, False)
```

Our first function, `forward(i)` , will turn on fwdright and fwdleft pins and then wait for the value of `i` , our argument before turning the motors off. On to our second function:

```
def right(i):
        GPIO.output(revright, True)
        GPIO.output(fwdleft, True)
        time.sleep(i)
        GPIO.output(revright, False)
        GPIO.output(fwdleft, False)
```

Our second function, `right(i)`, spins our robot on the spot in a clockwise direction for the duration provided as the argument `(i)`. To turn right we set the right motor to reverse and the left motor to forwards, wait for the user-defined number of seconds and then turn off the motors.

For our left and reverse functions you can refer to the full code at **http://bit.ly/LXF203_Robot**.

The last section of code is a try and except test:

```
try:
        print("R E A D Y")
except KeyboardInterrupt:
        print("E X I T")
        GPIO.cleanup()
```

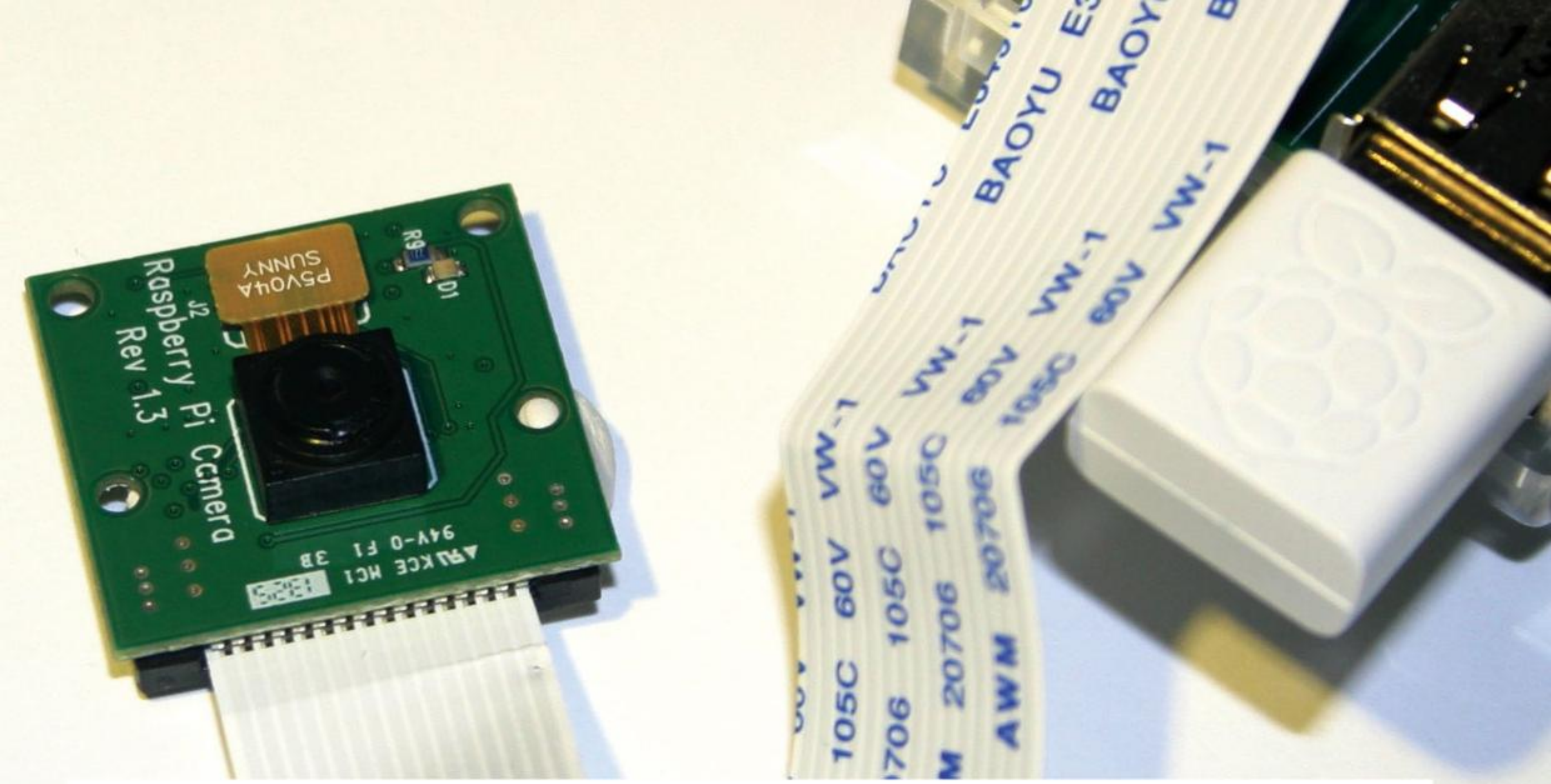
This will print R E A D Y when the code is executed, but if we press Ctrl+c it will print E X I T and then clean up the GPIO pins ready for use by another project.

Save your code as **robot.py** but we won't be running the code, rather we will now create a new file and save it as **test.py** in the same directory as **robot.py**.

Next, we'll import our **robot.py** code and use the functions inside of it to control our robot.

```
import robot
robot.forward(1)
robot.right(2)
robot.left(2)
robot.reverse(1)
```

Save the code and click Run > Run Module to test. Remember to pick up the robot before pressing Return or you'll have to chase after it! ■



Get started with the Pi Camera module

Mayank Sharma introduces you to the Pi Camera module and explains how you can use it to shoot away to your heart's content

The official Raspberry Pi camera module is fun little gadget, much like the board it complements. The camera module is a Full HD camera that plugs into the Raspberry Pi via the Camera Serial Interface (next to the Ethernet port) on the device. The camera has a fixed focus lens and 5 megapixel sensor. It can shoot still images with a maximum resolution of 2592×1944 as well as Full HD 1080p video at 30 frames per second, 720p video at 60fps and 640x480 at up to 90fps. And you get all this in a module that measures just 25x20x9mm in size and weighs a minuscule 3 grams!

To attach the camera, locate the Camera Serial Interface on the Raspberry Pi and pull the tab gently up. Now push the camera module's ribbon cable into the slot, with the silver contacts on the cable facing away from the Ethernet port. Remember not to push the cable in very deep. Now hold it in place with one hand and push the CSI tab back down with the other hand to lock the camera's ribbon.

With the hardware in place, it's now time to set up the software. Boot into Raspbian and log in. Before enabling the camera, make sure you refresh its repositories with the following:

```
sudo apt-get update
```

Then install any available updates with:

```
sudo apt-get upgrade
```

Once Raspbian is up to date, launch the Raspberry Pi's configuration script with the following command to tweak its settings:

```
sudo raspi-config
```

Scroll down the list to the Enable Camera option to make the Pi aware of the newly connected peripheral. Then exit the utility and restart the Pi.

Trigger happy

When it boots up, fire up a Terminal inside the graphical desktop to test the camera using the two command-line utilities **raspistill** and **raspivid** to capture still images and videos respectively. The command `raspistill -o image.jpg` displays a preview from the camera for 5 seconds and then takes a picture, which is saved as "image.jpg" in the current folder. You can use the `-t` switch to specify a different wait period in milliseconds. For example, `-t 20000` will display the preview for 20 seconds.

Similarly, `raspivid -d` will display a five-second video demo. You can define the size and location for the preview window using the `--preview` option – for example `raspivid --preview 0,0,1024,768` will display the preview in a 1024x768 window at the top-left corner of the screen. Use `--fullscreen` or `-f` to force the preview window to use the whole screen. The following command will capture 10 seconds of video:

```
raspivid -t 10000 -o video.mp4
```

Both raspistill and raspivid have quite a few options to help you take more interesting image and video captures. The --sharpness, --contrast and --brightness options can influence the output of the image. The camera also supports several exposure modes such as auto, night and sports that can be specified with the --exposure option. Similarly, the various white balance options including sun, cloud and shade can be specified with the --awb option.

Use the --width and --height option with raspistill to set the dimensions of the captured image. You can also tweak the compression level of the jpeg image by specifying a quality level between 0 and 100 with the --quality option.

The --width and --height options can also be used with the raspivid tool. It also uses the --bitrate option to control the quality of the video. To capture a Full HD video at 15 Mbits per second use --bitrate 15000000. Another related option is --framerate, which can accept a value between 2 and 30. To pause video during a capture, start it with the --keypress option. You can then press the Enter key to pause capturing and resume by pressing the Enter key once again. Press the X key followed by the Enter key to stop recording.

You can take advantage of the portable nature of the Raspberry Pi and its camera module to capture time-lapse video of things that are too subtle for the human eye to perceive, such as the movement of stars or the blossoming of flowers. A time-lapse video is composed of several images captured over several hours but played back at a much faster pace. For example, you can capture an image every 10 seconds for 6 hours and then string the set of images together in a time-lapse video at 24 frames per second. This reduces the 6-hour capture to a 30-second clip.

Jump through time

It doesn't take much work to set up the Pi for time-lapse photography. The following block of code will capture a shot every 30 second for the next 6 hours (21600000 milliseconds):

```
$ mkdir ~/images
$ cd ~/images
$ raspistill -o image-%04d.jpg -t 21600000 -tl 30000 &
```

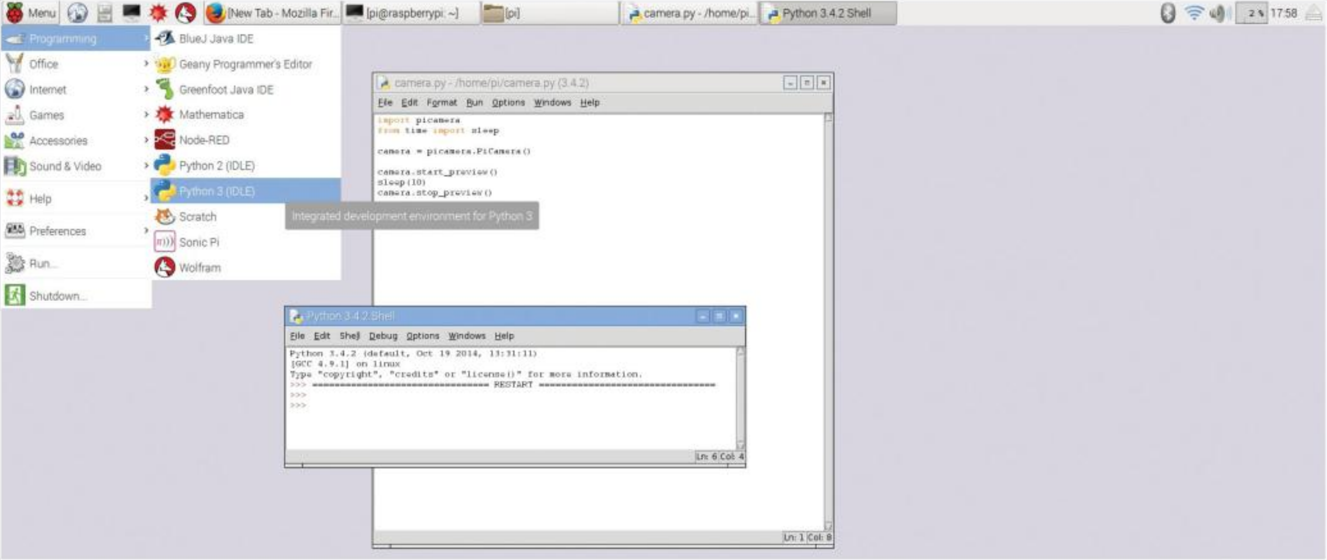
You can change the duration (-t option) or the gap between the shots (-tl option) by specifying different durations in milliseconds. The %04 instructs raspistill to save each image in a file called "image-*nnnn*.jpg' where *nnnn* is a four-digit

»

Pi bites

The camera module's tiny size makes it ideal for projects that require a small, high-resolution camera such as surveillance.

CONTROL THE CAMERA MODULE FROM PYTHON


Menu
[New Tab - Mozilla Fir...
[pi@raspberrypi: ~]
[pi]
camera.py - /home/pi...
Python 3.4.2 Shell
17:58
Programming
Office
Internet
Games
Accessories
Sound & Video
Help
Preferences
Run...
Shutdown...
BlueJ Java IDE
Geany Programmer's Editor
Greenfoot Java IDE
Mathematica
Node-RED
Python 2 (IDLE)
Python 3 (IDLE)
Scratch
Sonic Pi
Wolfram
Integrated development environment for Python 3
camera.py - /home/pi/camera.py (3.4.2)
File Edit Format Run Options Windows Help
Ln: 1 Col: 8
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Ln: 6 Col: 4

```
import picamera
from time import sleep

camera = picamera.PiCamera()

camera.start_preview()
sleep(10)
camera.stop_preview()
```

```
Python 3.4.2 (default, Oct 19 2014, 13:31:11)
[GCC 4.9.1] on linux
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
>>>
```

Head to Menu > Programming to use the IDLE editor to write your Python scripts.

Python's PiCamera library helps you control the camera module from within Python. You can install it on any distro with the following command:

```
sudo apt-get install python-picamera
```

Or if you plan to use Python 3:

```
sudo apt-get install python3-picamera
```

Once it's installed you can use the library to capture images. Here's a basic script that will display a preview for 10 seconds before capturing an image:

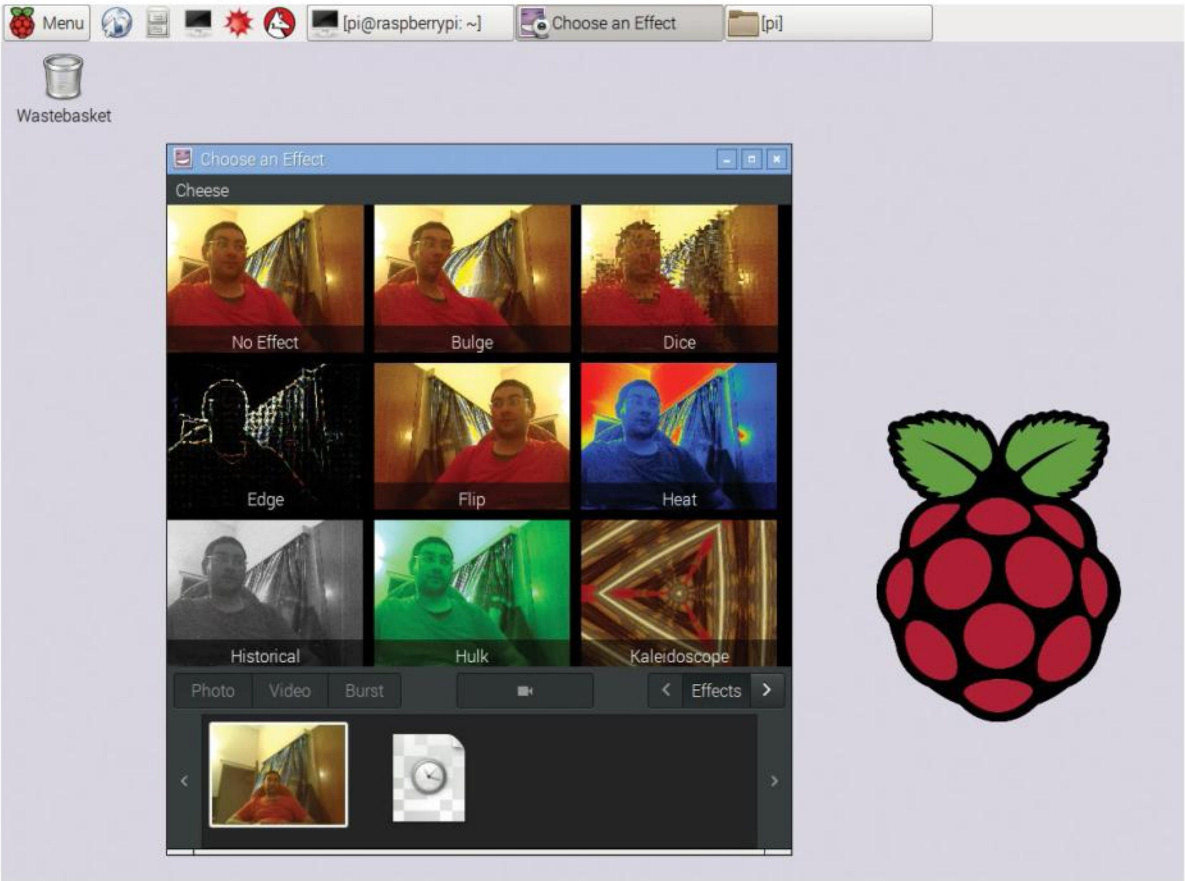
```
import picamera
from time import sleep

camera = picamera.PiCamera()

camera.start_preview()
sleep(10)
camera.capture('/home/pi/image.jpg')
camera.stop_preview()
```

Similarly, you can record 5 seconds of video with the following:

```
import picamera
from time import sleep

camera = picamera.PiCamera()
camera.start_recording('video.h264')
sleep(5)
camera.stop_recording()
```

You can also capture time lapse video using the capture_continuous() method of the picamera library.

The picamera library includes a number of useful modules such as picamera.encoders, picamera.streams, picamera.color and picamera.exec, among others. Refer to the official documentation at **http://picamera.readthedocs.io/en/release-1.12/** for more details and examples on the various modules.


Menu
[pi@raspberrypi: ~]
Choose an Effect
[pi]
Wastebasket
Choose an Effect
Cheese
No Effect
Bulge
Dice
Edge
Flip
Heat
Historical
Hulk
Kaleidoscope
Photo
Video
Burst
Effects

› Run `sudo modprobe bcm2835-v4l2` to access the camera board on **/dev/video0**. Then get the kernel to insert the module on boot automatically with `echo "bcm2835-v4l2" | tee -a /etc/modules`

number incremented with each successive image. The `&` sign at the end of the command instructs the Bash shell to run the command in the background. After six hours you'll have 720 images in the folder named image-0001.jpg, image-0002.jpg, and so on, up to image-0721.jpg.

You can now compile these individual images into a video. First, save them all in a text file:

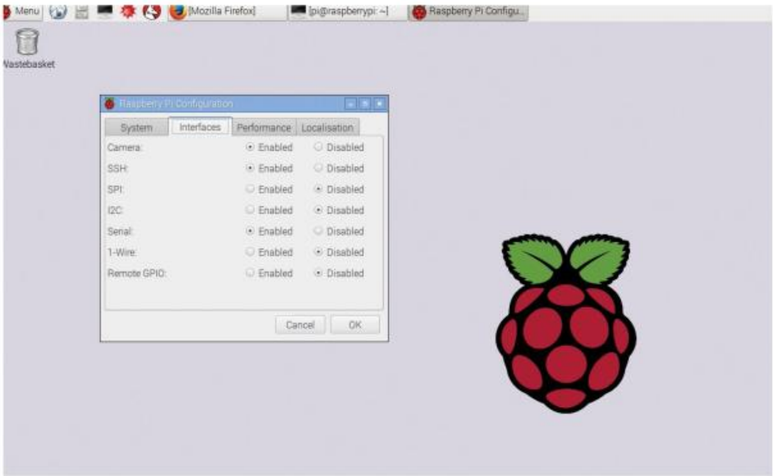
```
$ cd ~/images
$ ls *.jpg > allimages.txt
```

Then install the mencoder transcoding tool with:

```
sudo apt-get install mencoder
```

Once it's installed, compile the video with:

```
$ mencoder -ovc lavc -lavcopts vcodec=mpeg4:
aspect=16/9:vbitrate=8100000 -vf scale=1920:1080
-nosound  -o timelapse.avi -mf type=jpeg:fps=24
mf://@allimages.txt
```


Menu
[Mozilla Firefox]
[pi@raspberrypi: ~]
Raspberry Pi Configu...
Wastebasket
Raspberry Pi Configuration
System
Interfaces
Performance
Localisation
Cancel
OK

Camera: ● Enabled ○ Disabled
SSH: ● Enabled ○ Disabled
SPI: ○ Enabled ● Disabled
I2C: ○ Enabled ● Disabled
Serial: ● Enabled ○ Disabled
1-Wire: ○ Enabled ● Disabled
Remote GPIO: ○ Enabled ● Disabled

› You can also enable the camera from within Raspbian's desktop. Head to Menu > Preferences > Raspberry Pi Configuration > Interfaces and toggle the Enable radio button.

In this rather long command we've instructed mencoder to create a video using the mpeg4 codec from the open source libavcodec library and without any sound since we haven't captured any. The scale specifies the Full HD (1920x1080) resolution of the video, and the aspect specifies the widescreen aspect ratio of the video.

There's also a fair bit of maths involved. The bit rate is calculated using the formula (40 * 25 * width * height / 256), which comes out to 8100000. Then we set the frame rate of the resulting video to 24fps. Given that an individual frame (or one image) represents 30 seconds of real time, the real time represented by one second of our time-lapsed video comes to (24s * 30s) = 720s, which is equal to 12 minutes or one-fifth of an hour. In other words, five seconds of the time-lapsed video represents one hour of real time. Once you've got your head around the maths, tweak the options for the raspistill capture and the mencoder encoding until you get to a desired ratio between the real time and the video time.

Depending on the images it has to process, the mencoder command can take some time to complete. When it's done, you'll have a file named timelapse.avi inside the current directory. You can copy it to another computer using the `scp` command. Assuming the IP address of the Raspberry Pi is 192.168.1.2, head to another Linux computer on the network and enter:

```
$ scp pi@192.168.1.2:/home/pi/images/timelapse.
avi ~/Videos
```

This command will pull the video from the Pi to the Videos directory on your regular distro. ■

Turn any speakers into Wi-Fi speakers

Mayank Sharma explains how you can stream media from iOS and Android devices to your Raspberry Pi, to play it on speakers plugged into the Pi

These days, if you're like us, you've probably got your music all over the place – on cloud sharing services like Dropbox and Google Music as well as on local NAS drives. You can stream music from all these to Wi-Fi-enabled speakers that cost a bomb. Or, you can use a Raspberry Pi and transform any plain ol' speakers into AirPlay-compatible wireless speakers.

For this project, you'll need a pair of hi-fi speakers that you can connect to the Raspberry Pi via an Auxiliary (AUX) audio cable. If you are a true audiophile, you might scoff at the idea of using the Raspberry Pi to listen to music. But that's because you haven't yet tried the Volumio distro, which is designed especially for music lovers and is fully equipped to work with your hi-fi gear.

Using Volumio you can set up a headless media server on the Pi that can play music in all the popular formats from a connected USB device or NAS devices just as easily as it can stream Internet radio. What's more, the distro supports a large number of USB DACs, so you can plug in your amplifiers and other equipment to enhance your listening experience.

PLAY MUSIC FROM A CONNECTED USB DEVICE, NAS, OR AN APPLE OR ANDROID DEVICE VIA WI-FI

The best thing about Volumio is that it includes support for AirPlay devices out of the box. This means you can use the speakers connected to the Pi as remote speakers for your AirPlay-compatible Apple devices as well as Android devices with apps such as AirAudio that are available for free from the Google Play Store. Moreover, you can control Volumio from any other remote computer or device attached to your local network. This enables you to then select and queue music that'll play on the locally attached speakers.

Pump up the amps

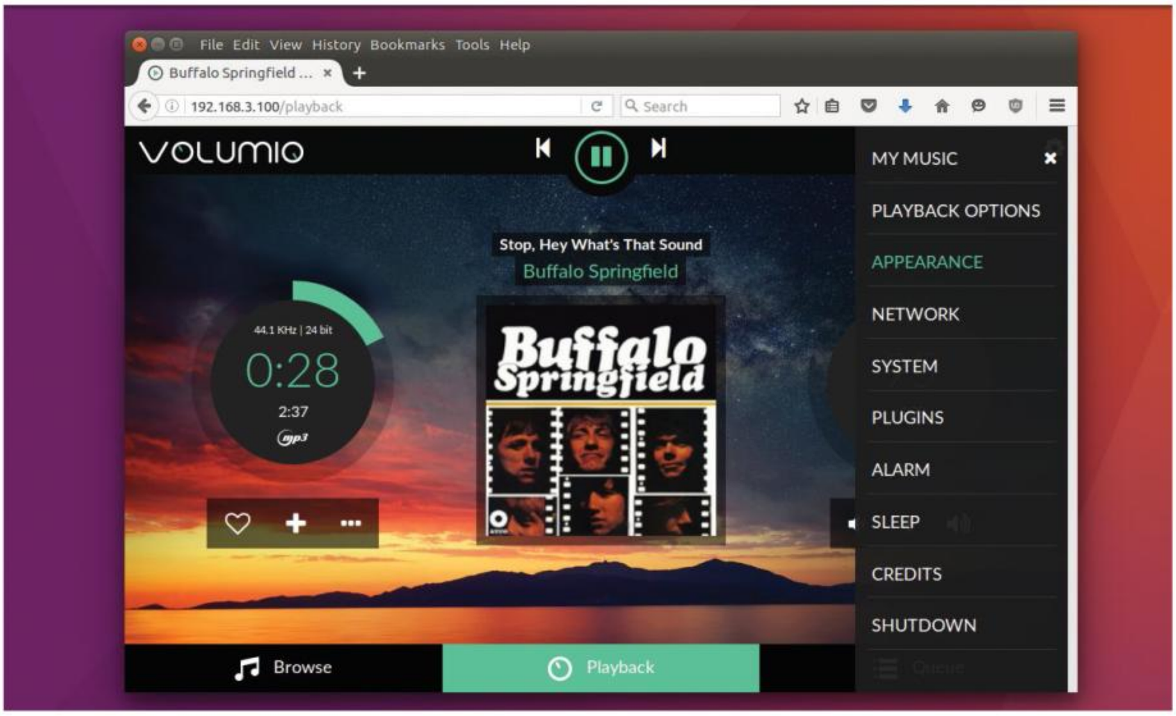
To get started, download the latest version of the Volumio distribution for the Raspberry Pi from its website at **https://volumio.org**. The image for the Pi is distributed in a Zip file, so you'll have to extract it before transferring it to the SD card you use with your Pi. Note that since the extracted image weighs almost 3GB in size, you'll need an SD card that's at least 4GB or more. It's also a good idea to use a Class 6 or higher card. The class number represents the speed of the card – the higher the value, the faster the card.

To write the Volumio image file in Windows, grab the USB Image Tool from **http://www.alexpage.de/usb-image-tool/download/**. You don't have to install the tool, just download the Zip file and

»

Pi bites

Although the Raspberry Pi does have USB ports, it's advisable to connect any USB devices using a powered USB hub to ensure that the devices get enough power.


File Edit View History Bookmarks Tools Help
Buffalo Springfield ...
192.168.3.100/playback
Search
VOLUMIO
Stop, Hey What's That Sound
Buffalo Springfield
44.1 KHz | 24 bit
0:28
2:37
mp3
Buffalo Springfield
MY MUSIC
PLAYBACK OPTIONS
APPEARANCE
NETWORK
SYSTEM
PLUGINS
ALARM
SLEEP
CREDITS
SHUTDOWN
Browse
Playback
Queue

› To improve your playback experience, tweak the settings available under Settings > Playback Options in Volumio's browser-based interface.

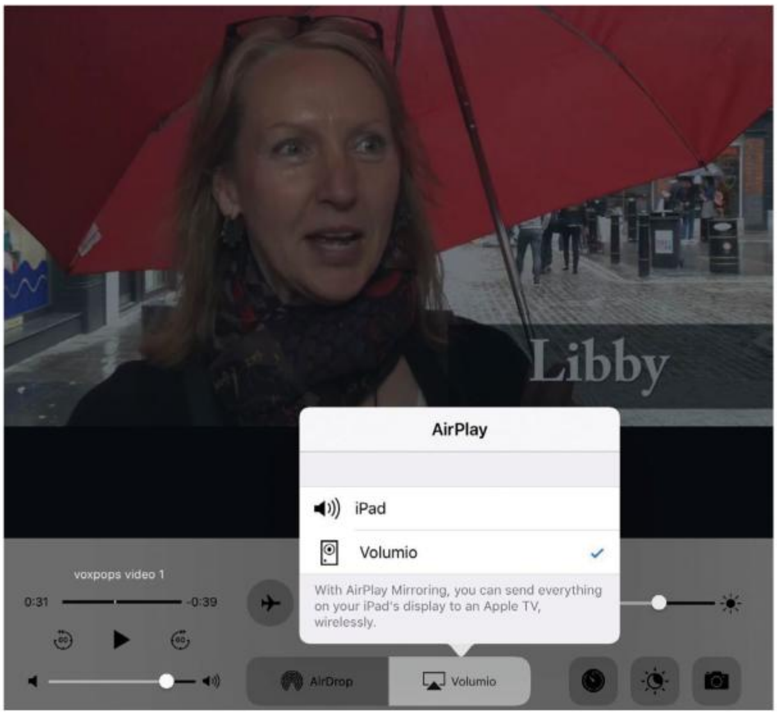
extract its contents. Now right-click on the USB Image Tool.exe file and select Run as Administrator from the contextual menu. The app will list your card in the left-hand pane. Make sure the tool is in Device Mode by selecting this from the pull-down menu in the top-left corner. To write the image to the card, click the Restore button and navigate to Volumio's extracted .img file. That's it. The image will now be transferred to the card – although the process may take some time, depending on the speed of the card.

If you're using Linux, you can use the venerable **dd** command to write Volumio's image to your SD card. Most distros will automatically mount the card when you connect it. You can find where the card is connected using the mount command, which will list all mounted devices. Assuming your card is mounted at **/dev/mmcblk0p1**, write the image to the card (rather than a partition inside it) with the following:

```
$ sudo umount /dev/mmcblk0p1
$ sudo dd bs=4M if=volumio-0.979-2016-08-20-pi.img of=/dev/mmcblk0
```

It'll take some time to write the image to the card. Once it's done, insert the card into the Raspberry Pi. Now power up the Pi and wait for about a minute to allow Volumio to boot up completely. That's all there's to it. Your AirPlay-compatible speakers are now ready. Grab any iOS device and make sure it's connected to the same Wi-Fi network as the Raspberry Pi. Then swipe up from the foot of the screen to open Control Centre. Tap AirPlay, which will bring up a list of AirPlay devices. Tap the entry labelled Volumio to stream audio from this iOS device to the speakers connected to the Raspberry Pi.

AirPlay isn't the only wireless audio streaming protocol out there, so don't feel left out if you don't have any iOS devices to stream music. The other popular protocol for streaming audio that's also supported by Volumio out-of-the-box is DLNA, and you can stream music from your Android devices to the Raspberry Pi over DLNA with the BubbleUPnP app (see *Stream Music with BubbleUPnP* on the opposite page).


Libby
AirPlay
iPad
Volumio ✓
voxpops video 1
0:31
-0:39
AirDrop
Volumio

With AirPlay Mirroring, you can send everything on your iPad's display to an Apple TV, wirelessly.

› It's easy to stream from an iOS device, but the secret sauce that powers Volumio is the venerable Music Player Daemon (MPD) music player server.

Local media library

In addition to streaming music from remote devices, you can also use Volumio to play back music from several other sources. It can read files stored on the same SD card or a locally attached USB device and even from another NFS or Samba share on the network. You can configure and control this and other aspects of Volumio via its browser-based interface from any computer on the same network as the Raspberry Pi. Before you can bring up Volumio's interface, you'll first have to find out the IP address assigned to the Raspberry Pi by visiting your router's administration page. Equipped with this address, head over to another computer on the same network and enter the IP address assigned to the Raspberry Pi.

Volumio's web interface looks very appealing and the controls are very intuitive. But before you can use it to play music, you'll need to point it to your library. If you've got music files on a USB drive, Volumio will automatically import them all as soon as you plug in the drive. On the other hand, if you want to point Volumio to a network share that houses your music, click the gears icon in the top-right corner of the main screen to open Volumio's configuration menu and click My Music. Click the Add New Drive button and fill in the IP address of the network share, along with the path to the folder that contains the music files.

Once you've added your music, head back to the main Volumio interface and click on the Browse button in the bottom left of the screen. This will bring up a list of the available music sources, which includes a bunch of Internet radio stations as well as any connected USB disks and network drives. Browse through the devices, pick a track and the song you select will pipe through the speakers connected to the Pi. You can access the Volumio interface from any computer on the network to queue and control playback on the remote speakers.

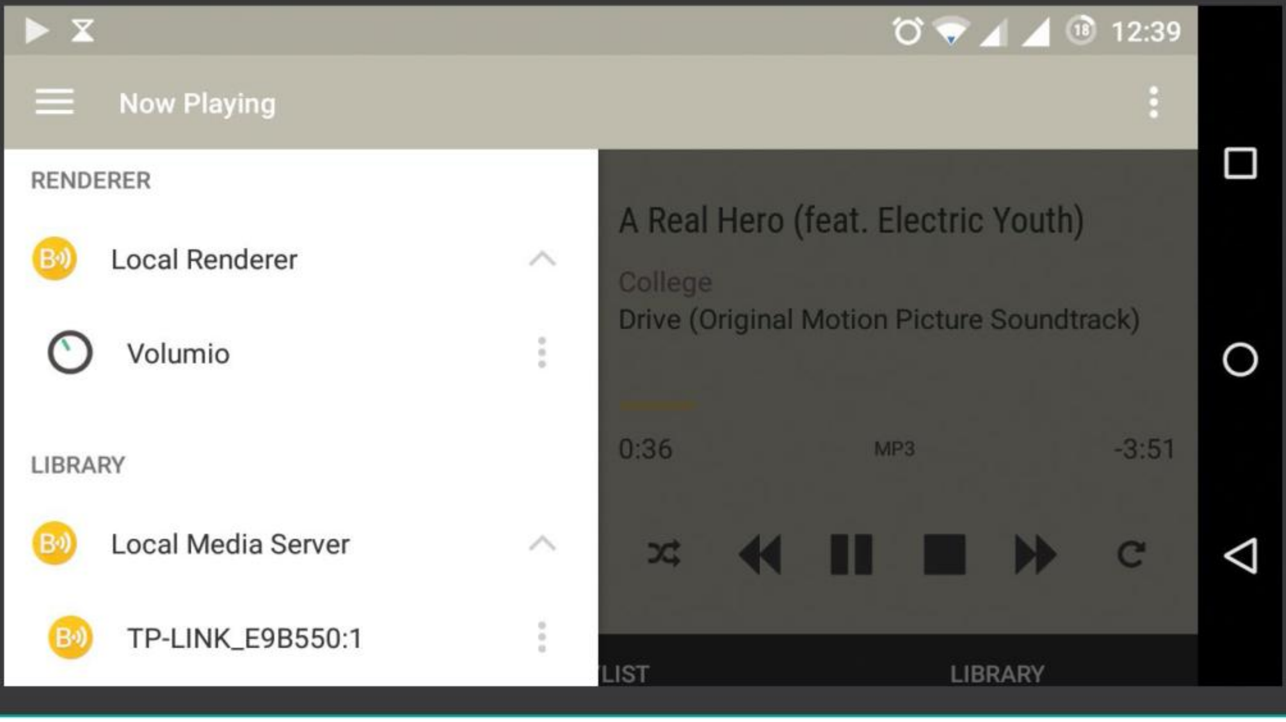
What's more, you can also ask Volumio to fetch music from the Internet. As we've mentioned, the Browse button automatically gives you access to a handful of online radio stations. (Want more? Turn the page for more about internet radio.) You can also point Volumio to your Spotify Premium account to play music from the popular service. For this, head to Settings > Plugins and install the Spotify plugin to use the service. Also, in addition to the browser-based interface, you can control playback with one of the several Volumio apps available on the Android Play Store. ■

STREAM MUSIC WITH BUBBLEUPNP

In addition to AirPlay, the Volumio distro for Raspberry Pi also supports the DLNA protocol for streaming music. On non-iOS devices such as Android, you can use the BubbleUPnP app, a wonderful tool that can stream music to your Volumio-equipped Raspberry Pi via the DLNA protocol. BubbleUPnP is available as an ad-supported free download on the Google Play Store.

Once BubbleUPnP is installed, tap the three parallel lines in the top-left corner to open the menu. By default the app is configured to play music on the local device. To make it stream the music to the Pi, expand the list of Renderers and tap on the Volumio entry. That's all there is to it. Now switch to the Library tab in the main app window and browse and select any music on the device, which BubbleUPnP will them stream to the Pi.

Besides music stored locally on the Android device, BubbleUPnP can also stream music from cloud-based services. To set up the device to stream content from the cloud libraries, tap the menu button and under Library, expand the Cloud section to view a list of supported cloud services including Google Music, Google Drive, Dropbox, OneDrive and more. Tap the name of the service you wish to add and enter your login credentials when prompted. Repeat the process to add other services as well. When you're done, tap the name of the service you've added to browse its contents. Tap to select any music, and BubbleUPnP will pipe it through the remote speakers attached to your Pi.



BubbleUPnP can also stream music from USB disks attached to the USB port on modern Wi-Fi routers, which are usually all DLNA-certified.

Build an internet radio from a Pi

Nate Drake shows you how to set up your Pi so that you can enjoy internet radio wirelessly anywhere in the house... or even beyond

One of the most fun and customisable projects available for the Pi owner is to turn it into a wireless device capable of tuning in to internet radio stations. Internet radio is a thing apart from regular radio stations. As the name suggests, internet radio stations stream over the internet rather than broadcasting over the airwaves.

Of course there's no reason why radio lovers can't simply crank up their AM/FM handset and listen to radio in the old-fashioned way. But internet radio offers many advantages. The chief among these must be audio quality: the signal strength and error checks put in place by internet communication protocols offers much crisper and clearer audio than most household radio sets. Because radio stations are transmitting over an internet connection, range is no longer an issue either, and you can tune into stations around the world wherever you have a network connection.

This is especially important for people living away from their home country. British expats living in continental Europe, for instance, have difficulty receiving the BBC World Service on a shortwave radio, but an internet radio can stream it without any issue. Certain countries also jam certain shortwave radio frequencies, but it is more difficult to jam internet broadcasts.

A quick search online will show that a number of people have turned a Pi into an internet radio receiver in many different and inventive ways. For this reason, we've included the basic information here to get you started but leave it to you to decide how you wish to customise it.

› **Feel free to be creative with this project. This vintage radio has had a Pi installed as well as a small RGB display.**

Radio preparations

As you'll see from the next few pages, the actual code required to set up an internet radio on your Raspberry Pi is only a few lines long. Before you get stuck in, however, it's worth taking some time to consider your particular setup.

Your first consideration should be your choice of Raspberry Pi. The Raspberry Pi 3 is the most expensive model but it does have a 3.5mm audio jack, which will be compatible with most earphones and speakers. It also has several USB ports if you want to use a set of USB speakers, which can be obtained very cheaply online. Most screens including the Official Raspberry Pi Touch Screen are also designed to work with the Pi 3.

The Raspberry Pi Zero in many ways is perfect for this project as very little processing power is required. Its small size also means it can fit nicely inside an existing radio or small case. It also is much less expensive than a Pi 3. This said, the Zero is not compatible with every external screen and has no audio jack, so you will have to use USB speakers or output your audio over an HDMI cable, for example, to a television.

Once you've decided on your Pi model and speakers, you may wish to consider a case for the radio. Technically you could just connect your Pi via HDMI to a television and listen to internet radio stations that way, but this isn't the most elegant way to do things.

Some stylish Pi users have enthusiastically disassembled vintage radios, removed the innards except for the speaker and placed their Pi inside for that rustic feel, sometimes complete with an RGB screen such as Adafruit's handy PiPlate, which has an RGB display and buttons you can program with favourite radio stations. See **www.adafruit.com/products/1110** for information.

If you choose to go down this road, make sure you have a basic knowledge of DIY and are comfortable with a soldering iron as some assembly may be required.

Should you wish to go for a more fully featured Sony Walkman feel for your Pi, you may prefer to use a regular case in combination with a small touchscreen such as Adafruit's 2.8-inch PiTFT. Visit **www.adafruit.com/product/1601** to see the display in action.

You may prefer to do something more unusual such as make the casing out of Lego (or similar interlocking-block construction products) or even cardboard. It isn't particularly important which

»

Adafruit i2c 16x2 RGB LCD Pi Plate
6 CHNS Halifax
classic rock
SELECT

The Pi Plate comes with five keypad buttons which can be programmed to move between stations

PLAYING RADIO WITH YMPD

To get started, either visit the ympd website on the Pi to download the software (**www.ympd.org**) or connect to your Pi via SSH and type this command, then press Return:

```
wget https://www.ympd.org/downloads/ympd-1.2.3-armhf.tar.bz2
```

Extract the app with the command:

```
tar -xvf ympd-1.2.3-armhf.tar.bz2 && cd ympd
```

Now move the ympd program to your applications directory by running:

```
sudo mv ympd /usr/bin
```

Finally start ympd running with:

```
sudo ympd --webport 80
```

To access the ympd interface open your web browser either on the Pi or another device connected to your network and go to **http://yourpiIPaddress:80** – for example, **http://192.168.1.118:80**.

The interface is very simple and allows you to browse your music collection as well as play individual stations, such as BBC Radio Four.

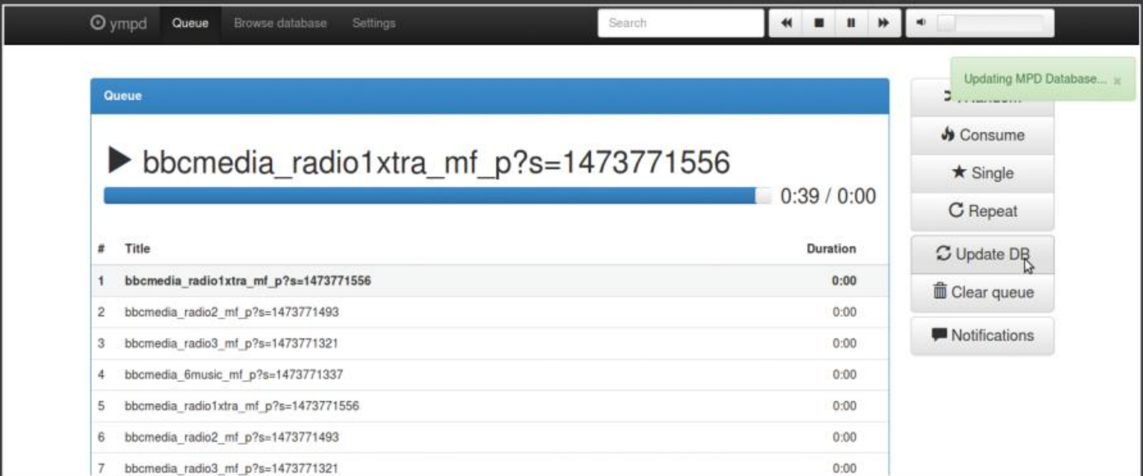
If you like ympd and want it to start up each time Pi boots, you can do this by editing the configuration files on your Pi. First enter this command:

```
sudo nano /etc/rc.local
```

Then use the arrow keys to scroll to the blank line just above the text "exit 0". Paste in the following:

```
sleep 30; sudo ympd --webport 80
```

Press Ctrl +X, then Y, then return to save your changes.



Queue

▶ bbcmedia_radio1xtra_mf_p?s=1473771556

0:39 / 0:00

#	Title	Duration
1	**bbcmedia_radio1xtra_mf_p?s=1473771556**	**0:00**
2	bbcmedia_radio2_mf_p?s=1473771493	0:00
3	bbcmedia_radio3_mf_p?s=1473771321	0:00
4	bbcmedia_6music_mf_p?s=1473771337	0:00
5	bbcmedia_radio1xtra_mf_p?s=1473771556	0:00
6	bbcmedia_radio2_mf_p?s=1473771493	0:00
7	bbcmedia_radio3_mf_p?s=1473771321	0:00

Ympd can be accessed on any web browser. Simply click on a radio station to begin playing. If you've added new stations – we show you how over the page – click "Update DB" to load them.

Pi bites

If you want a portable radio, consider buying a "Pi Borg" from the Pi Hut at **http://bit.ly/PiBorg**. This enables you to power your Pi with regular AA batteries.

80s: The Knack - My
Sharona (1979)

Another suggestion for your radio setup. Here the Pi and wiring are inside a clip lock lunchbox. Note that a USB soundcard has been added (left) for better audio.

you choose so long as you plan the layout carefully, leaving adequate room for the power cable and speakers.

The Music Player Daemon

The heart of the internet radio lies in the handy program mpd (Music Player Daemon). Combined with its handy command line tool **mpc**, it can be used to add radio stations, as well as skip back and forth between them.

There are of course a number of programs capable of playing internet radio such as the Pi's built in web browser, but none of them can be guaranteed to work with every setup. More importantly, it's possible that you'll be using your Pi radio without an attached monitor, so it's very useful to be able to start and stop the radio by connecting via SSH.

If you do want to use software to interact with the music player daemon, make sure to test it out from the command line so you can be sure the basic code is working.

The Music Player Daemon runs in the background, so if you have to restart the Pi, then it will start playing again once the desktop loads. It is possible to disable this feature (see below).

Some basic commands include:

`mpc current` Returns the name of the radio station that's currently playing.

`mpc play` Instructs mpd to play the currently selected track/station. You can add a number to play the equivalent station in your current playlist – for example if you've added BBC Radio 1, 2 and 3 in that order then `mpc play 3` would tell mpd to start playing BBC Radio 3.

`mpc move` Changes the order in which stations are played – for example `mpc move 1 2` moves a station *from* position 1 *to* position 2.

`mpc pause` Instructs mpd to pause the current track. This is very handy if the track has started playing automatically when you don't want it to.

`mpc next` Instructs mpd to play the next track.

`mpc prev` Instructs mpd to play the previous track.

`mpc playlist` Lists all songs in your current playlist.

MPC RADIO
Streaming Internet Radio
80% online
stopped
Press PLAY or REFRESH
adafruit
RASPBERRY PI

Mr UK Tech Reviews has a customised Python window specially designed for playing radio on Adafruit's mini display.

Say yes to ympd

While using mpc from the command line is very effective, it can be a little tedious to type out commands each time you want to play a track. This is why there are a number of graphical front-ends for mpd.

Our recommended player of choice is **ympd** as it's very easy to set up. The program runs in the background and allows you to view your list of radio stations through a web browser, either on the Pi itself if you have a screen, or another device.

Follow the steps in *Playing radio with ympd* on the previous page to get started. If you're accessing ympd via another device like a laptop computer, remember that the sound from the radio itself will still play through your Pi. If you want to listen to internet radio on that device, you will need to set up separate software on there.

Advanced setup

As we've mentioned, internet radio setups are often customised, and among the most common customisations that people like to put into practice are either adding a small display to show the track now playing, or some physical buttons to switch stations.

This involves making use of the Pi's GPIO (General Purpose Input/Output) pins, which are used to connect hardware components. An excellent video tutorial showing how to use the Radio Pi plate, with links to source code, is available via the channel of a YouTuber called "Usual Panic" at **youtu.be/HPLh8-IVJx4**.

YouTuber Steven Yoshida also provides an excellent video summary of the parts for his internet radio using Adafruit's 2.8-inch touchscreen display at **www.youtube.com/watch?v=Vw4_S3sb8Is**.

The full steps for setting up the small screen and connecting it to the Pi are available via the Adafruit website at **http://bit.ly/Pi-TFT**.

In order to make his radio portable, Steven Yoshida makes use of a small battery pack for the Pi. Otherwise you're tied to your power cable.

While we're on the subject of power, you might also find it handy to have a way to mute the radio volume. If you're using a touchscreen display it's easy to adjust the volume manually. Alternatively you could consider replacing your regular USB power cable with one with a switch, such as those sold by Pi Hut.

Crossed wires

Just as there are a number of ways to set up your project, there are equally a myriad of ways that things can go wrong. It's important before you set up your radio to test that the sound is working properly with your speakers of choice before trying to listen to any streams.

If you have more than one audio output – say you've connected earphones to the Pi's earphone jack but are also connected to a television by HDMI – then you may need to right-click the volume button at the top right of the screen to select the output you want.

If you're connecting over SSH and can't use the Pi's desktop to change your audio settings, then simply type `mpc outputs` to see a list of available audio devices. The list will be numbered in a form such as "Output 1 (My Alsa Device)". Use the command `mpc enable #` – for example `mpc enable 2` – to enable the speakers of your choice.

If mpd becomes unresponsive when you're using it, in the first instance try simply connecting via SSH or opening Terminal and running the command `sudo service mpd restart`. If you find such problems are happening regularly, it may be best to stop the mpd background daemon and just run it manually with these two commands:

```
sudo service mpd stop
sudo mpd --no-daemon --stdout --verbose
```

Keep an eye on the Terminal window – the "--verbose" flag will force mpd to show any error messages. If you're still having troubles, try visiting the helpful mpd troubleshooting page at **http://mpd.wikia.com/wiki/Music_Player_Daemon_HOWTO_Troubleshoot**.

If you have any other software running on the Pi that connects to the internet, you may find that you aren't able to connect to ympd on Port 80 as outlined in *Playing radio with ympd* on the previous page. If you are unable to load the ympd window, first stop the program with the command `sudo killall ympd` and then launch it again using a different port – for example `sudo ympd –webport 94`. You should then be able to access ympd by opening a web browser on your network and going to **http://yourpiIPaddress:94**.

If you decide to use the small TFT touchscreen from Adafruit, you may find it tricky to navigate the Pi's smaller screen. You can get around this by »

Pi bites

If you do choose to use USB speakers, they may require more power than your Pi's USB ports can supply. If so, use a powered USB hub which connects to mains electricity.

ADDING STATIONS

The Pi is capable of playing any internet radio station. The basic format for adding new stations using the mpc command line utility for the Music Player Daemon is `mpc add LINK` –for example `mpc add http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio1_f_p?s=1473771325&e=1473785725&h=01fae51e24dd4d26dc181f593a94defb`.

This is quick and easy (if long-winded) but you may struggle at first to find valid links for stations, particularly as the URLs will change from time to time.

Stephen Phillips has also written an excellent script on his website (**http://blog.scphillips.com/posts/2014/05/bbc-radio-on-the-raspberry-pi-v2/**) which can automatically fetch the current links for BBC Radio. At press time these are:

BBC Radio 1 : **http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio1_mf_p?s=1473771325&e=1473785725&h=01fae51e24dd4d26dc181f593a94defb**

BBC Radio 2 : **http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio2_mf_p?s=1473771493&e=1473785893&h=b8c1c754cb6cc971fed2ca5b19e97d67**

BBC Radio 3 : **http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio3_mf_p?s=1473771321&e=1473785721&h=c7d8a753d2663d9f4630cfcbb7e1d1ae**

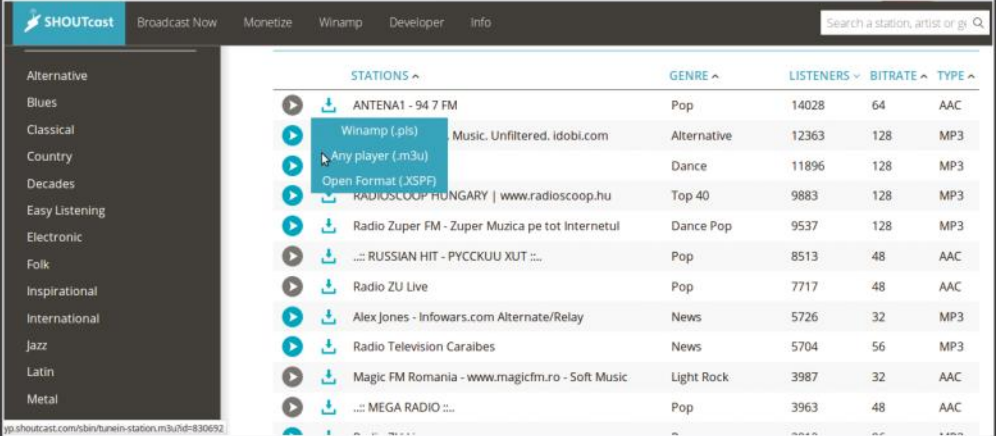
BBC Radio 4 : **http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio4extra_mf_q?s=1473771502&e=1473785902&h=6c94d3cedf0a32e28dd4e8bd33e5fe44**

BBC Radio 5 Live : **http://open.live.bbc.co.uk/mediaselector/5/redir/version/2.0/mediaset/http-icy-mp3-a-stream/proto/http/vpid/bbc_radio_five_live**

BBC World Service: **http://bbcwssc.ic.llnwd.net/stream/bbcwssc_mp1_ws-eieuk_backup**

ShoutCast (**www.shoutcast.com**) has links to hundreds of radio stations which can either be played directly on the website or downloaded as a playlist. This can cause issues for mpd, which needs a link to work with. Fortunately the website **gebbl.net** has an excellent script you can download to convert the playlist files to an actual web stream. See **https://www.gebbl.net/2013/10/playing-internet-radio-streams-mpdmpc-little-bash-python/** for more information.

If you want to categorise your radio stations rather than have them all in one big list, just add the streams you want and then save as a playlist –for example `mpc save bbc`. A playlist file will appear in the location **/var/lib/mpd/playlists**. You can load the playlist again by using `mpc load` – for example `mpc load bbc`. You can then either play it using the command `mpc play` or using your graphical program of choice such as ympd.



STATIONS	GENRE	LISTENERS	BITRATE	TYPE
ANTENA1 - 94 7 FM	Pop	14028	64	AAC
Music. Unfiltered. idobi.com	Alternative	12363	128	MP3
	Dance	11896	128	MP3
RADIOSCOOP HUNGARY | www.radioscoop.hu	Top 40	9883	128	MP3
Radio Zuper FM - Zuper Muzica pe tot Internetul	Dance Pop	9537	128	MP3
..:: RUSSIAN HIT - PYCCKUU XUT ::..	Pop	8513	48	AAC
Radio ZU Live	Pop	7717	48	AAC
Alex Jones - Infowars.com Alternate/Relay	News	5726	32	MP3
Radio Television Caraibes	News	5704	56	MP3
Magic FM Romania - www.magicfm.ro - Soft Music	Light Rock	3987	32	AAC
..:: MEGA RADIO ::..	Pop	3963	48	AAC

Websites like ShoutCast have downloadable playlists for radio stations. However, you'll need an actual link for the stations you want in order to be able to listen to them using mpc.

TO PLAY DOWNLOADED MUSIC AS WELL AS RADIO STATIONS, JUST TELL MDP WHERE IT'S LOCATED

using a larger screen such as the Official Raspberry Pi 7-inch Touchscreen Display (**https://thepihut.com/products/official-raspberry-pi-7-touchscreen-display**), which is a good deal larger.

The Adafruit website also links to a tutorial video by YouTuber "Mr UK Tech Reviews," who designed a special window in Python complete with handy buttons specifically for people using the Raspberry Pi as an internet radio with Adafruit's display. See **http://bit.ly/Raspberry-Radio** for more information.

If on the other hand your radio is working perfectly, you might want to use it to play your downloaded music as well as radio stations. In order to do this, you only need to tell mdp where your music is located. Simply edit the mdp configuration file with the following command:

```
etc/mpd.conf
```

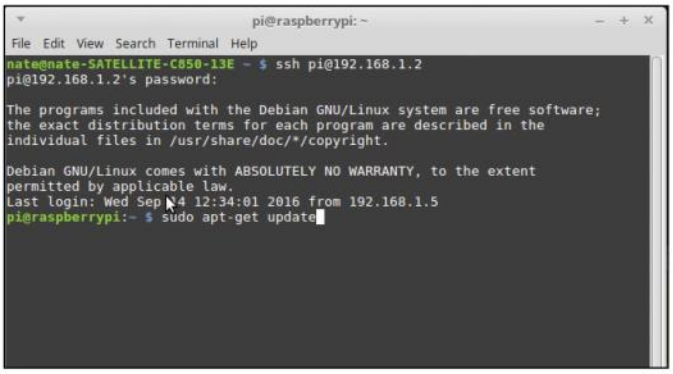
Scroll down to the words "music_directory". Replace the path **/var/lib/mpd/music** with the actual location of your music files – for example something like **/home/pi/Music**. Press Ctrl + X, then Y, then Return to save and exit. Next run the command `mpc update` for mpc to detect your music. The program can recognise and play MP3 files as well as Apple's AAC files.

Once this is done and you are happy with your setup, we encourage you to share your project online so others can benefit from your experience. The Instructables website (**www.instructables.com/about/submit.jsp**) has a number of Pi Radio projects already but if you feel yours stands out, please share it your fellow Pi users! ■

Pi bites

For a full list of possible mpc commands type `mpc help` in the Terminal.

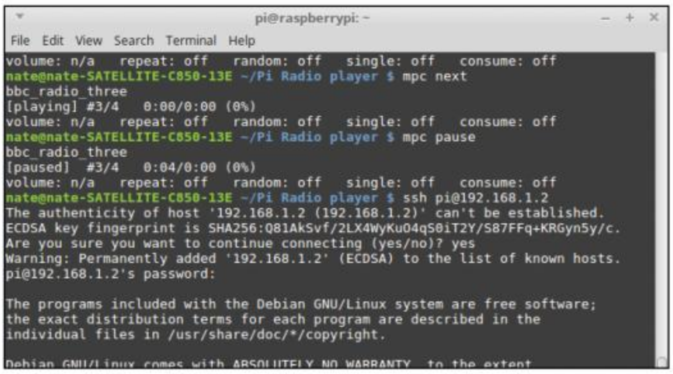
TURN YOUR PI INTO A MUSIC PLAYER



```
nate@nate-SATELLITE-C850-13E ~ $ ssh pi@192.168.1.2
pi@192.168.1.2's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Sep 14 12:34:01 2016 from 192.168.1.5
pi@raspberrypi:~ $ sudo apt-get update
```

1 Set up your Pi

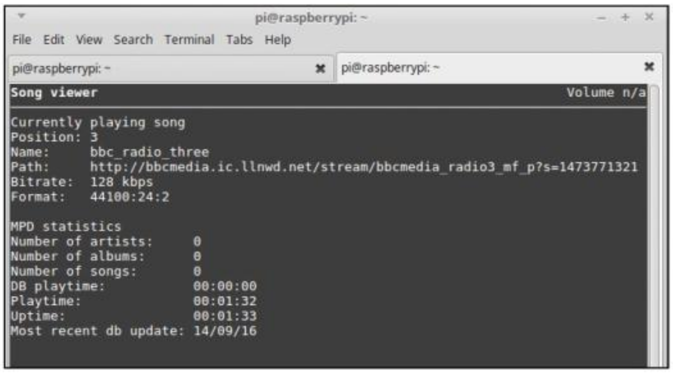
Make sure your Pi is plugged into power and you've connected any speakers you want to use. Open your Pi's Terminal or connect via SSH, then type the command `sudo apt-get update` and press Return to prepare the Pi to install the software.



```
volume: n/a   repeat: off   random: off   single: off   consume: off
nate@nate-SATELLITE-C850-13E ~/Pi Radio player $ mpc next
bbc_radio_three
[playing] #3/4   0:00/0:00 (0%)
volume: n/a   repeat: off   random: off   single: off   consume: off
nate@nate-SATELLITE-C850-13E ~/Pi Radio player $ mpc pause
bbc_radio_three
[paused]  #3/4   0:04/0:00 (0%)
volume: n/a   repeat: off   random: off   single: off   consume: off
nate@nate-SATELLITE-C850-13E ~/Pi Radio player $ ssh pi@192.168.1.2
The authenticity of host '192.168.1.2 (192.168.1.2)' can't be established.
ECDSA key fingerprint is SHA256:Q81AkSvf/2LX4WyKuO4qS0iT2Y/S87FFq+KRGyn5y/c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.2' (ECDSA) to the list of known hosts.
pi@192.168.1.2's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
```

2 Install Music Player Daemon

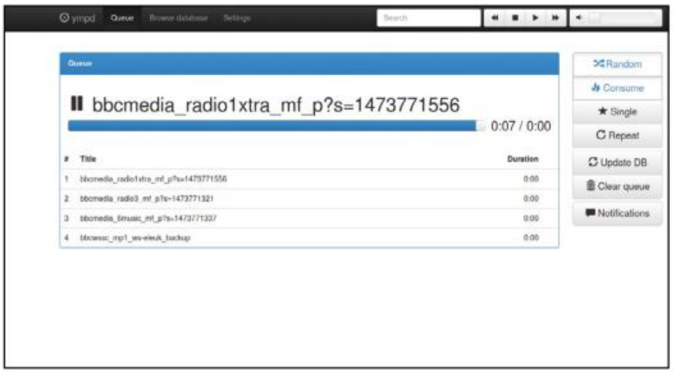
Enter the command `sudo apt-get install mpd mpc` to install the player software. The daemon will run at all times in the background. Use the **mpc** program to control it via the command line. Next add audio streams for your favourite radio stations – see *Adding Stations* on the previous page.



```
Song viewer                                                    Volume n/a

Currently playing song
Position: 3
Name:     bbc_radio_three
Path:     http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio3_mf_p?s=1473771321
Bitrate:  128 kbps
Format:   44100:24:2

MPD statistics
Number of artists:     0
Number of albums:      0
Number of songs:       0
DB playtime:           00:00:00
Playtime:              00:01:32
Uptime:                00:01:33
Most recent db update: 14/09/16
```

3 Play tracks/stations

Use the command line to test your setup by running `mpc play` to make sure that your audio streams are working. Use `mpc pause` to stop when you are satisfied that the audio is playing correctly. You might need to change the audio output on your Pi.



bbcmedia_radio1xtra_mf_p?s=1473771556

0:07 / 0:00

#	Title	Duration
1	bbcmedia_radio1xtra_mf_p?s=1473771556	0:00
2	bbcmedia_radio3_mf_p?s=1473771321	0:00
3	bbcmedia_6music_mf_p?s=1473771337	0:00
4	bbcwssc_mp1_ws-eieuk_backup	0:00

4 Add interface

This step is optional but recommended. Although you can add just play tracks from the command line, this isn't very intuitive. So use the browser based player **ympd** (see *Playing radio with ympd*, p75) or refer to the main article for information on more exotic options such as touchscreens.

SPECIAL 2016 AWARDS ISSUE!
PHONES | WEARABLES | AUDIO | TVS | CARS | HOME
T3
The Gadget Magazine
UPGRADE TODAY!
iPHONE 7 & WATCH 2
→ NEW APPLE KIT THAT REWRITES THE RULEBOOK
10TH ANNIVERSARY
GADGET of the YEAR
REVEALED: 2016'S TOP TECH IN EVERY CATEGORY THAT COUNTS
THE BEST OF THE BEST AS VOTED FOR BY YOU
ISSUE 261 AWARDS 2016
PRINTED IN THE UK £4.99
Future

ON SALE NOW!

AVAILABLE AT **WHSMITH, MYFAVOURITEMAGAZINES.CO.UK**
OR SIMPLY SEARCH FOR T3 IN YOUR DEVICE'S APP STORE

SUBSCRIBE TODAY AND SAVE! SEE WWW.MYFAVOURITEMAGAZINES.CO.UK/T3

Google play | Available on the App Store | zinio™

BECOME AN EXPERT CODER THE EASY WAY

GuruGuide
Nobody goes deeper into technology

CODING ACADEMY 2017

FULLY REVISED & UPDATED EDITION

LEARN TO CODE FAST TODAY!
• PYTHON • SWIFT • RUST • R STUDIO
• MARIA SQL • MONGO • REDIS

180 PAGES OF TUTORIALS
LEARN CORE CODING TECHNIQUES AND ADVANCED SKILLS

FREE
Digital edition of this book!
See page 178 for more information
iPad & iPhone only

Future
TGG14 2016
PRINTED IN THE UK
£12.99

OUT NOW! WITH FREE DIGITAL EDITION

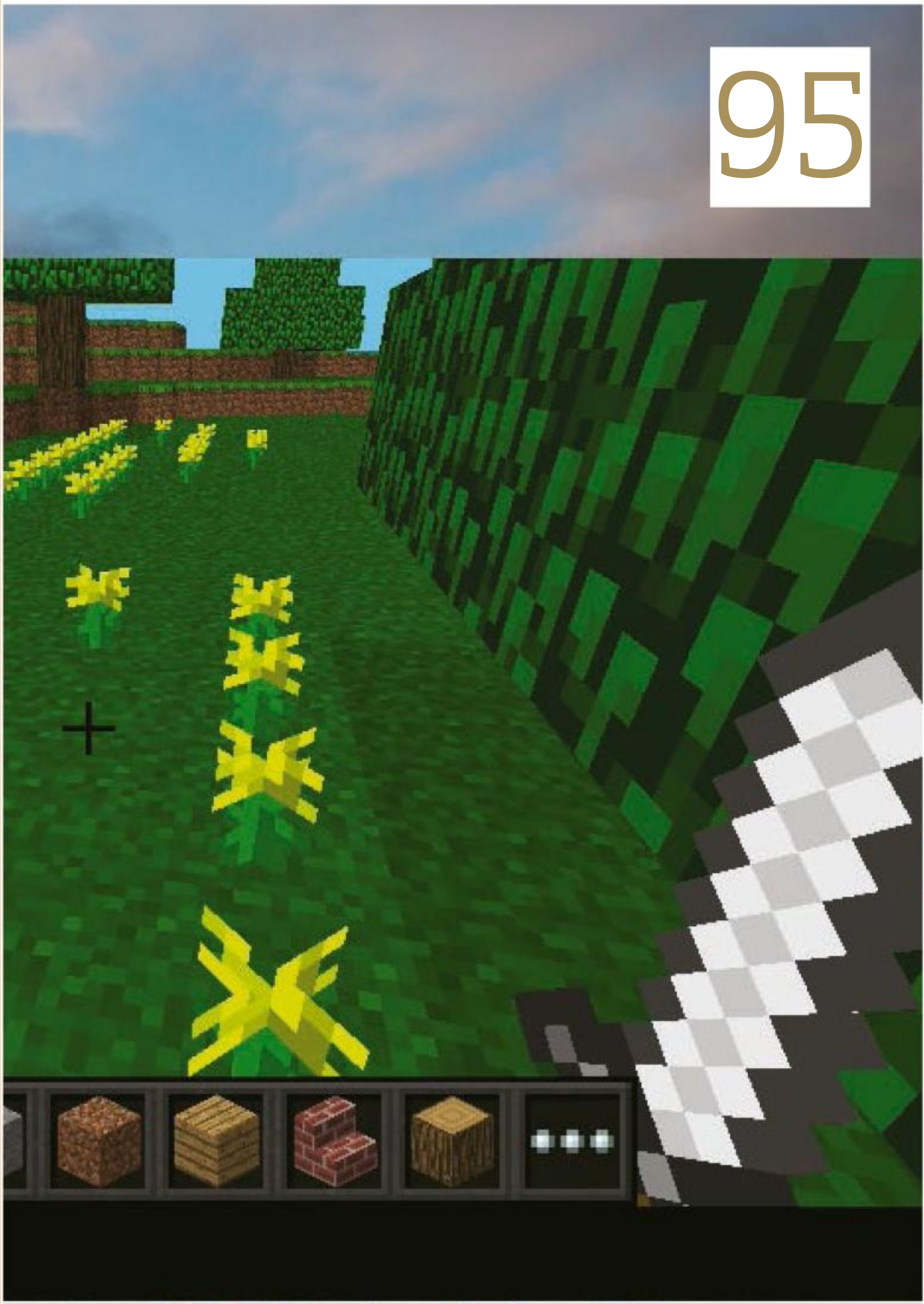
Future

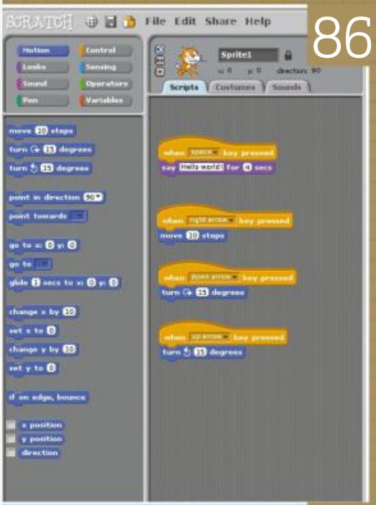
ORDER YOUR COPY TODAY!
Order online at
https://www.myfavouritemagazines.com
Also available in all good newsagents

Available on the App Store
GET THE MAGAZINE ON Google play
WHSmith

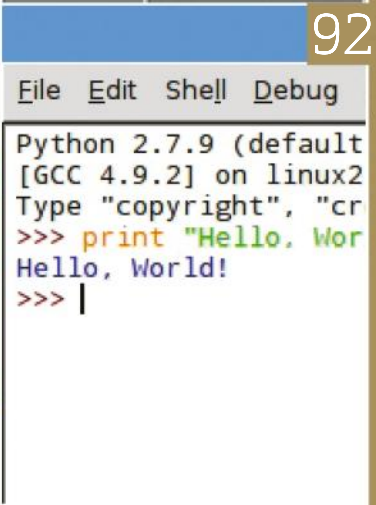
Pi CODE

Learn how to code in your language of choice




86
SCRATCH
File Edit Share Help
Motion
Looks
Sound
Pen
Control
Sensing
Operators
Variables
Sprite1
Scripts
Costumes
Sounds
move 10 steps
turn 15 degrees
turn 15 degrees
point in direction 90
point towards
go to x: 0 y: 0
go to
glide 1 secs to x: 0 y: 0
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
x position
y position
direction

```
when space key pressed
say Hello world! for 4 secs

when right arrow key pressed
move 10 steps

when down arrow key pressed
turn 15 degrees

when up arrow key pressed
turn 15 degrees
```


92
File Edit Shell Debug

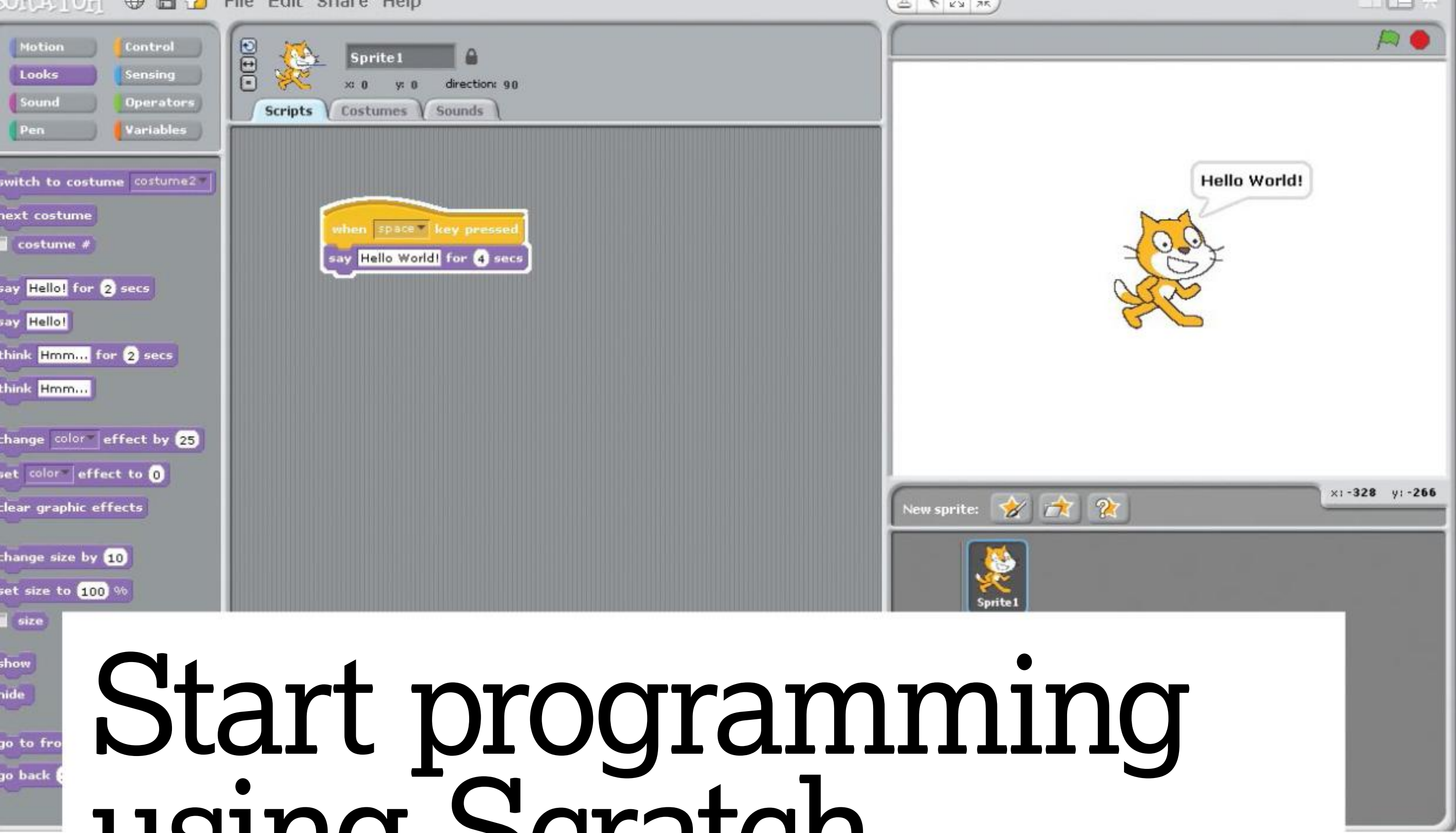
```
Python 2.7.9 (default
[GCC 4.9.2] on linux2
Type "copyright", "cr
>>> print "Hello, Wor
Hello, World!
>>> |
```

TURN EVERYTHING YOU TOUCH INTO GOLD WITH SOME ELEMENTARY CODING IN MINECRAFT

86 Get started with Scratch

92 Get to know Python coding

95 Begin coding in Minecraft


Motion
Looks
Sound
Pen
Control
Sensing
Operators
Variables
Sprite1
x: 0 y: 0 direction: 90
Scripts
Costumes
Sounds
switch to costume costume2
next costume
costume #
say Hello! for 2 secs
say Hello!
think Hmm... for 2 secs
think Hmm...
change color effect by 25
set color effect to 0
clear graphic effects
change size by 10
set size to 100 %
size
show
hide
when space key pressed
say Hello World! for 4 secs
Hello World!
New sprite:
x: -328 y: -266
Sprite1

Start programming using Scratch

Ben Everard shows you how to build a cat and mouse game using this straightforward beginner's programming language on the Raspberry Pi

There's a wide range of programming languages that can be used with the Raspberry Pi, but Scratch is great for beginners, because it introduces many of the concepts of programming while being easy to use. It's especially good for creating graphical programs such as games.

You'll find Scratch on the desktop in Raspbian, so there's no need to install anything – just click on the icon to get started. (Many other operating systems for the Pi don't support Scratch, although you can download Scratch for Windows, Mac or Linux at **https://scratch.mit.edu**. Our tutorials will be equally applicable whatever your platform.)

The main window in Scratch is split into three sections: the bits you can use to make programs are on the left; you piece them together in the middle; and the programs run in the area on the right. Each program is made up of a number of sprites (pictures) controlled by scripts which you assemble in the middle panel like building blocks.

At the top-left are eight colour-coded categories or functions. To begin, click on Control, and blocks of code will appear below. From these, drag **When Space Key Pressed** into the scripts panel in the middle. What do you want to happen when the space key is pressed? Click on Looks at top-left, and drag **Say Hello For 2 Secs** directly beneath the first piece. The two lock together into a script.

You'll notice that "Hello" and "2" have a different background, indicating they can be edited. Click on **Hello** and change this to **Hello World!**, then click on **2** and change this to **4**. Press Return to exit the text box. To run the script, press the space key. Congratulations, you've made a Scratch program!

Want to do more? Click on Control again, drag **When Space Key Pressed** into the scripts area, then click on "Space Key" and change this to **Right Arrow**. Click on Motion at top-left and drag **Move 10 Steps** beneath the Right Arrow block. Now the cat will take a walk if you hit the arrow key. Get the idea? Let's try something a bit more complex...

Variables and messages

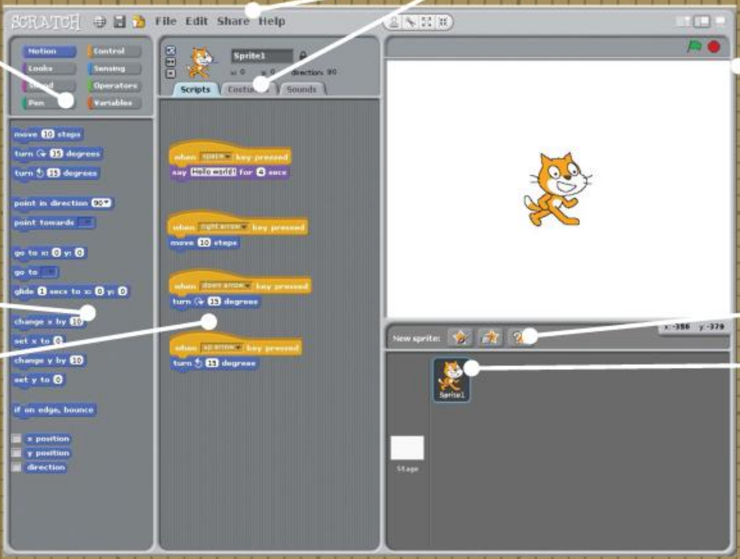
Sooner or later, you're going to want to get your program to remember something. It might be a number, a piece of text, or anything. You can do this using variables. These are little pieces of the computer's memory that your program can place pieces of data in. In step 5, we create a pair of these to store some numbers in, although we could also put text in them. Note that in some programming languages, you have to create different types of variables if you want to store different types of data, but you don't need to worry about that in Scratch.

Once you've created a variable, you can use it in a few ways. First, you have to set it to be a particular value, then you can use it to evaluate conditions (which we'll do in steps 8 and 13), or you can output it.

Messages

If you create a number of scripts, you might need to communicate between them. Sometimes you can do this with variables, but it is often better to use messages. These can be used to trigger scripts in the same way as keypresses can. When one script broadcasts a message, it will then trigger all the scripts that start with a matching **When I Receive...** Like variables, messages have names, so they have to be created first, and for a script to trigger it has to be linked to the same message as the broadcast.

EXPLORING SCRATCH


SCRATCH
File Edit Share Help
Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
Sprite1
Scripts
Costumes
Sounds
New sprite:
Stage

Categories Choose what type of code you want to use next. They're colour-coded to help you identify which type each piece of code is.

Options Menus and buttons along the top offer further options. Under Share you can upload your project to **scratch.mit.edu** to let the world see it.

Costumes/Sounds Click a tab to show the costumes (that is, images) and sounds for the current sprite.

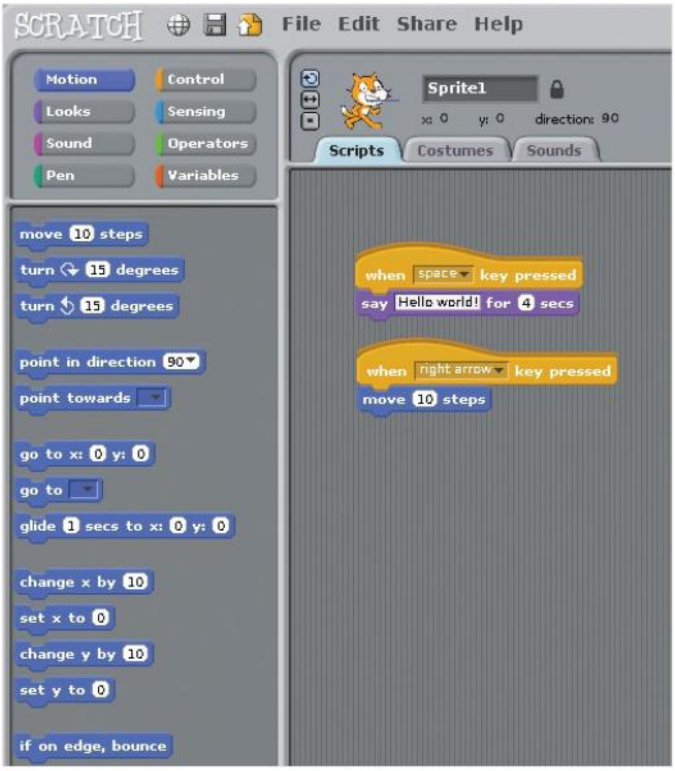
Canvas This is where sprites are displayed and their behaviour is previewed.

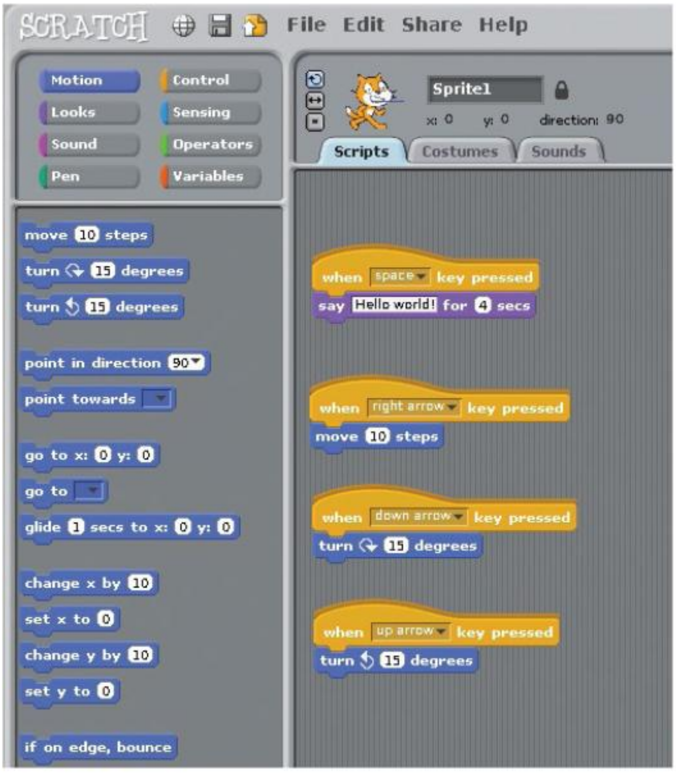
Code Pick a category and pieces of code appear here. Drag any you want into the scripts panel to build up your program.

New Sprite Here you can create a new sprite directly from an image file.

Scripts These are the scripts for the current sprite. You can edit various parameters. To remove a piece of code, drag it back into the left-hand panel.

Sprite select As you add more sprites, they appear here. Click one to select it, then click the tabs in the middle panel to see its scripts and costumes.


SCRATCH
File Edit Share Help
Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
Sprite1
x: 0 y: 0 direction: 90
Scripts
Costumes
Sounds
move 10 steps
turn ↻ 15 degrees
turn ↺ 15 degrees
point in direction 90
point towards
go to x: 0 y: 0
go to
glide 1 secs to x: 0 y: 0
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
when space key pressed
say Hello world! for 4 secs
when right arrow key pressed
move 10 steps

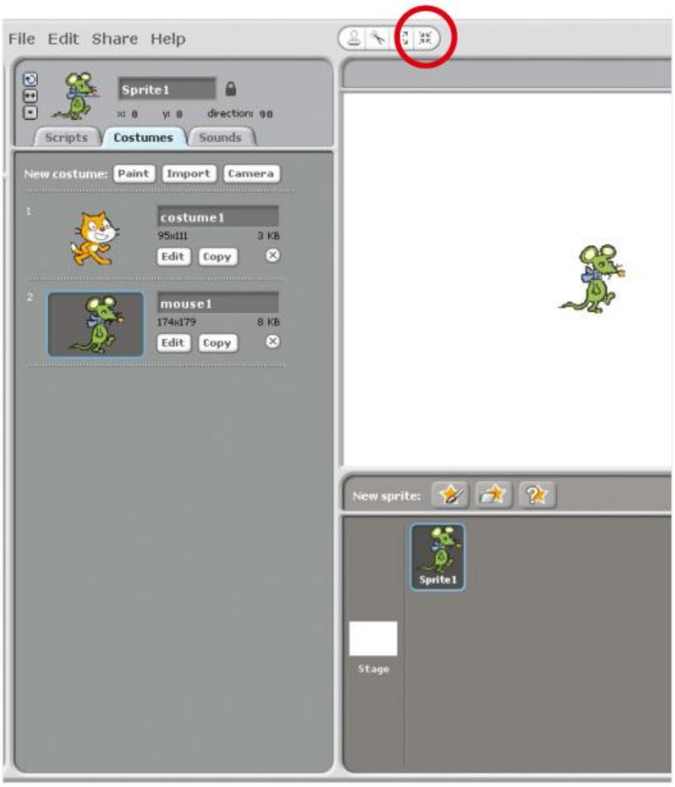

SCRATCH
File Edit Share Help
Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
Sprite1
x: 0 y: 0 direction: 90
Scripts
Costumes
Sounds
move 10 steps
turn ↻ 15 degrees
turn ↺ 15 degrees
point in direction 90
point towards
go to x: 0 y: 0
go to
glide 1 secs to x: 0 y: 0
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
when space key pressed
say Hello world! for 4 secs
when right arrow key pressed
move 10 steps
when down arrow key pressed
turn ↻ 15 degrees
when up arrow key pressed
turn ↺ 15 degrees

1 Create your scripts

We'll build on the trial program we've begun. At this point your scripts panel should look like this. (If not, follow the steps in the introductory text on the opposite page.) Now click on Control again, and drag *two* instances of **When Space Key Pressed** into the scripts panel...

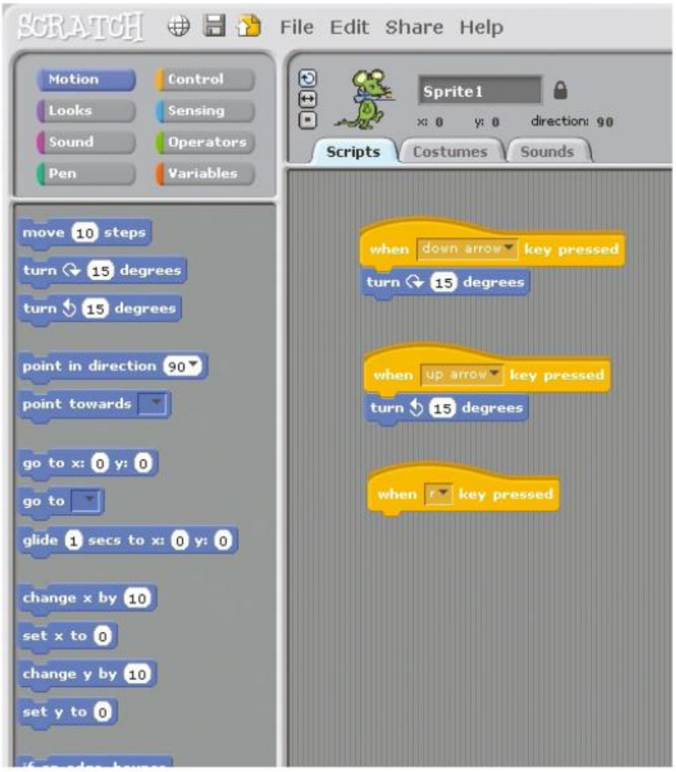
2 Add more scripts

In the first, click on "Space" and change the key to **Down Arrow**. Now click Motion again, and drag **Turn (clockwise) 15 Degrees** beneath this block. In the second **When ... Pressed** block, change "Space" to **Up Arrow**, then add **Turn (anti-clockwise) 15 Degrees** to this script. »


File Edit Share Help
Sprite1
Scripts
Costumes
Sounds
New costume: Paint Import Camera
costume1
mouse1
Edit Copy
New sprite:
Sprite1
Stage

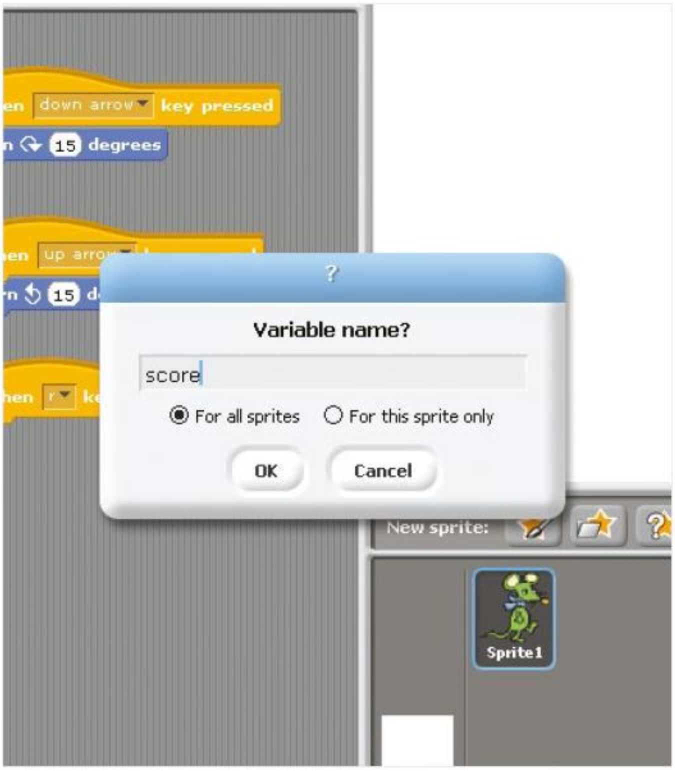
3 Create the mouse

Change the sprite image from the default cat to a mouse by clicking the Costumes tab, then Import > Animals > Mouse 1. Next reduce the sprite size by clicking on the Shrink Sprite icon (circled) and then the mouse. We set it to about the size of our thumbnail.


SCRATCH
File Edit Share Help
Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
move 10 steps
turn 15 degrees
turn 15 degrees
point in direction 90
point towards
go to x: 0 y: 0
go to
glide 1 secs to x: 0 y: 0
change x by 10
set x to 0
change y by 10
set y to 0
Sprite1
Scripts
Costumes
Sounds
when down arrow key pressed
turn 15 degrees
when up arrow key pressed
turn 15 degrees
when r key pressed

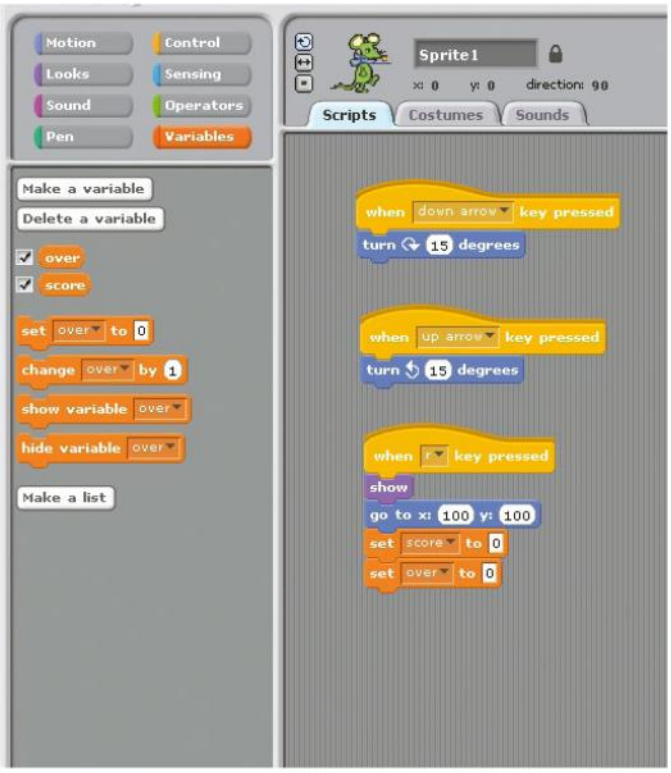
4 Set keys

Click the Scripts tab in the middle panel, and change **When Space Key Pressed** to **When r Key Pressed**. We'll use this to start a new game (**r** is for reset). Then drag **Say Hello World...** off the bottom of the script. If you drop it back in the left side, it will be deleted.


Variable name?
score
● For all sprites
○ For this sprite only
OK
Cancel
New sprite:
Sprite1

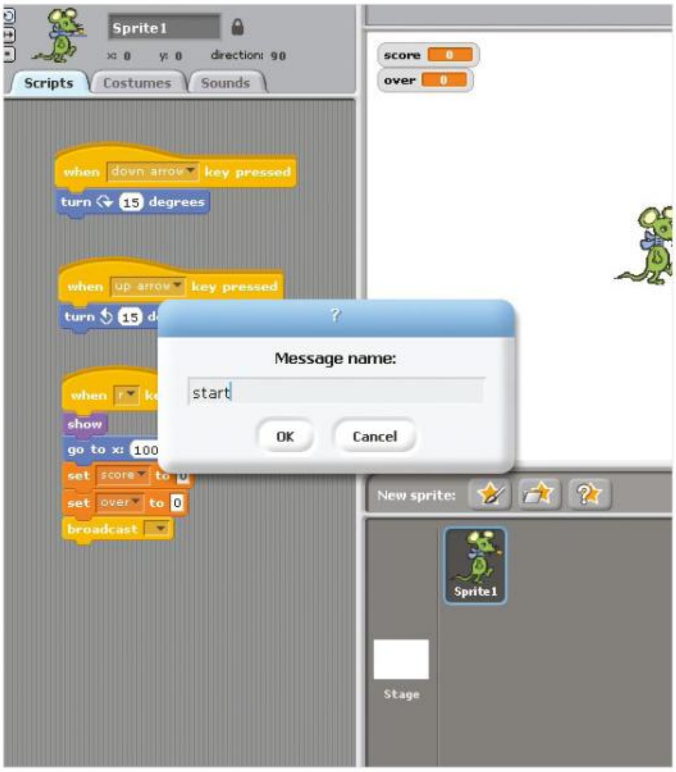
5 Create and name variable

Click on Variables in the top-left (see the box on the previous page for more details about what they are). Click on **Make A Variable** and enter the variable name as **score**. Repeat the process to create a second variable called **over**.


Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
Make a variable
Delete a variable
☑ over
☑ score
set over to 0
change over by 1
show variable over
hide variable over
Make a list
Sprite1
Scripts
Costumes
Sounds
when down arrow key pressed
turn 15 degrees
when up arrow key pressed
turn 15 degrees
when r key pressed
show
go to x: 100 y: 100
set score to 0
set over to 0

6 Reset the score

Click on Looks and drag **show** under the script **When r Key Pressed**. Now click on Motion and add **Go To X:100**, **Y:100** under the same script (don't forget to change 0s to 100s). Finally, from Variables, add both **Set Score To 0** and **Set Over To 0**.



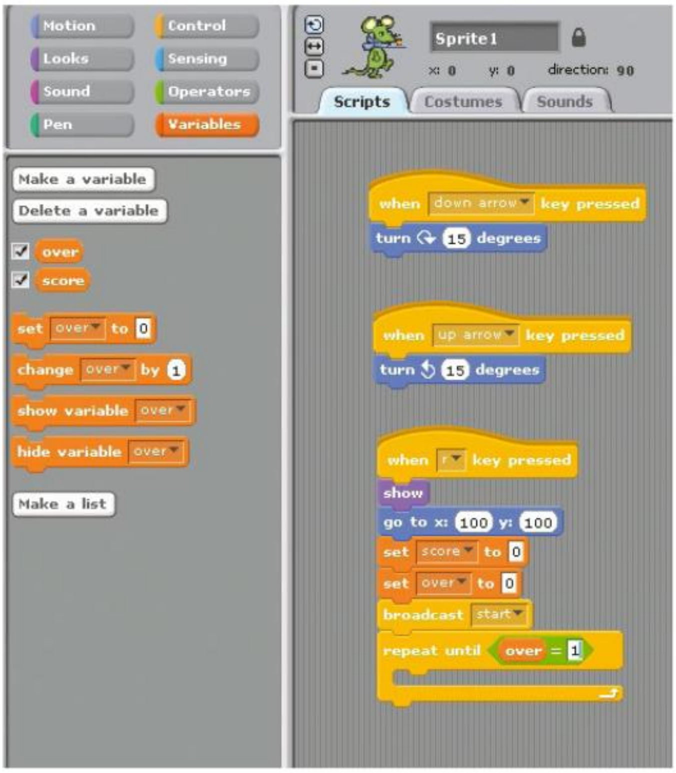
```
when [down arrow] key pressed
turn ↻ 15 degrees

when [up arrow] key pressed
turn ↺ 15 d

when [r] k
show
go to x: 100
set [score] to 0
set [over] to 0
broadcast [ ]
```

Message name: start

Sprite1
x: 0 y: 0 direction: 90
Scripts
Costumes
Sounds
score 0
over 0
?
OK
Cancel
New sprite:
Sprite1
Stage

7 Add broadcast

From Control, add the block **Broadcast...** to the bottom of the **When r Key Pressed** script. Once it's there, click on the drop-down menu and select 'New..'. and give the message the name **start**. We'll use this to let the other sprite know that the game has started.

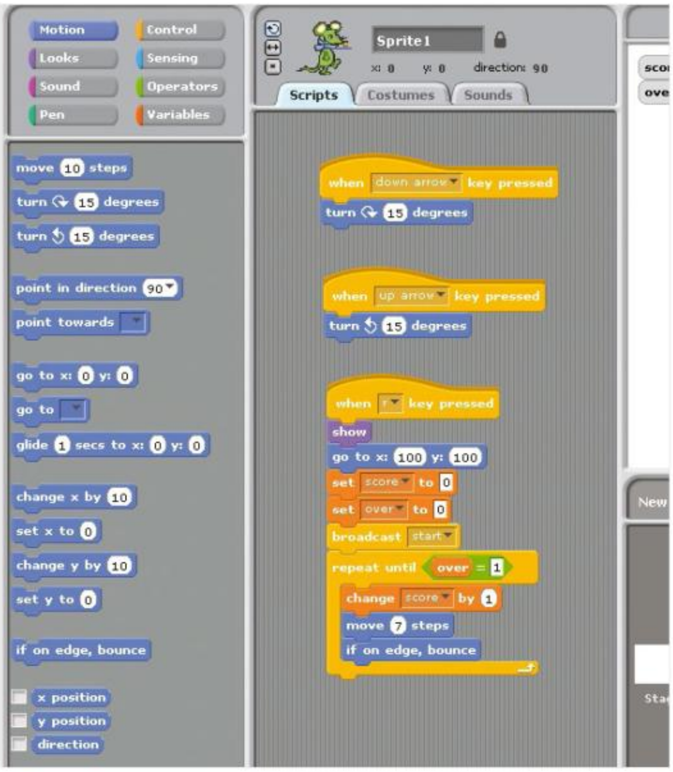


```
when [down arrow] key pressed
turn ↻ 15 degrees

when [up arrow] key pressed
turn ↺ 15 degrees

when [r] key pressed
show
go to x: 100 y: 100
set [score] to 0
set [over] to 0
broadcast [start]
repeat until <(over) = 1>
```

Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
Make a variable
Delete a variable
☑ over
☑ score
set over to 0
change over by 1
show variable over
hide variable over
Make a list
Sprite1
x: 0 y: 0 direction: 90
Scripts
Costumes
Sounds

8 Create a loop

We can create loops that cycle through the same code many times. Continue the script with **Repeat Until...** (from Control), and then drag and drop **... = ...** (from Operators), then drag **Over** (from Variables) into the left-hand side of the **=** and enter **1** on the right.

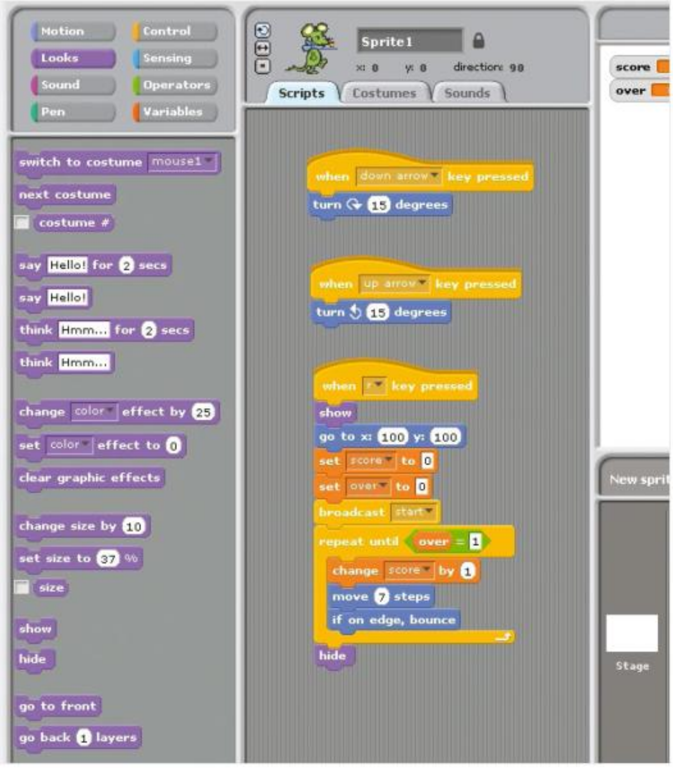


```
when [down arrow] key pressed
turn ↻ 15 degrees

when [up arrow] key pressed
turn ↺ 15 degrees

when [r] key pressed
show
go to x: 100 y: 100
set [score] to 0
set [over] to 0
broadcast [start]
repeat until <(over) = 1>
    change [score] by 1
    move 7 steps
    if on edge, bounce
```

Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
move 10 steps
turn ↻ 15 degrees
turn ↺ 15 degrees
point in direction 90
point towards
go to x: 0 y: 0
go to
glide 1 secs to x: 0 y: 0
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
x position
y position
direction
Sprite1
x: 0 y: 0 direction: 90
Scripts
Costumes
Sounds

9 Add to your loop

Inside the **Repeat Until Over = 1** block, add **Change score By 1** (from Variables), **Move 7 Steps** (from Motion) and **If On Edge, Bounce** (also from Motion). These three pieces of code will be constantly repeated until the variable **over** gets set to **1**.



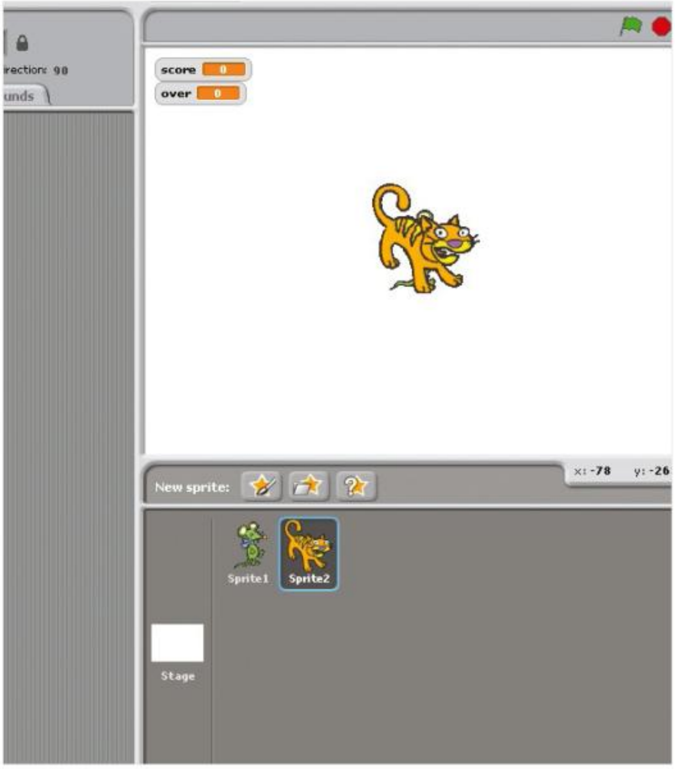
```
when [down arrow] key pressed
turn ↻ 15 degrees

when [up arrow] key pressed
turn ↺ 15 degrees

when [r] key pressed
show
go to x: 100 y: 100
set [score] to 0
set [over] to 0
broadcast [start]
repeat until <(over) = 1>
    change [score] by 1
    move 7 steps
    if on edge, bounce
hide
```

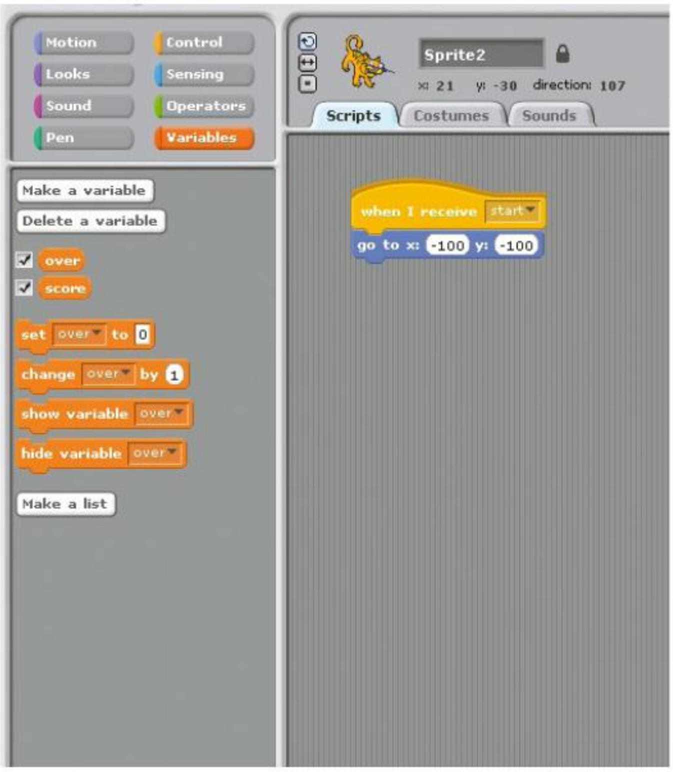
Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
switch to costume mouse1
next costume
costume #
say Hello! for 2 secs
say Hello!
think Hmm... for 2 secs
think Hmm...
change color effect by 25
set color effect to 0
clear graphic effects
change size by 10
set size to 37 %
size
show
hide
go to front
go back 1 layers
Sprite1
x: 0 y: 0 direction: 90
Scripts
Costumes
Sounds
score
over
Stage

10 Hide the mouse

Once the game has finished (and the cat has got the mouse), the **Repeat Until** loop will end and the program will continue underneath it. Drag **Hide** (from Looks) under the loop, so the mouse disappears when this happens.

»


direction: 90
score 0
over 0
New sprite:
x: -78 y: -26
Sprite1
Sprite2
Stage


Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
Make a variable
Delete a variable
☑ over
☑ score
set over to 0
change over by 1
show variable over
hide variable over
Make a list
Sprite2
x: 21 y: -30 direction: 107
Scripts
Costumes
Sounds

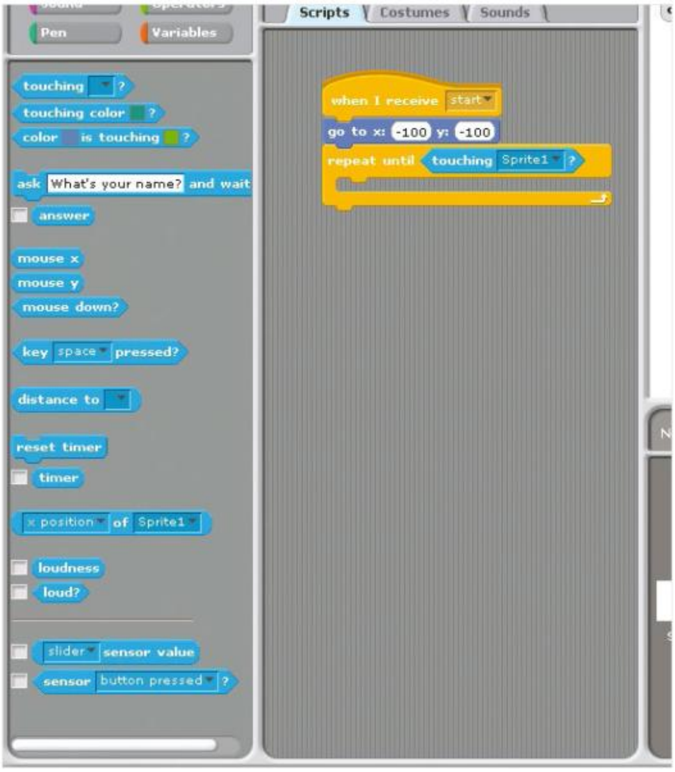
```
when I receive start
go to x: -100 y: -100
```

11 Add your cat and resize it

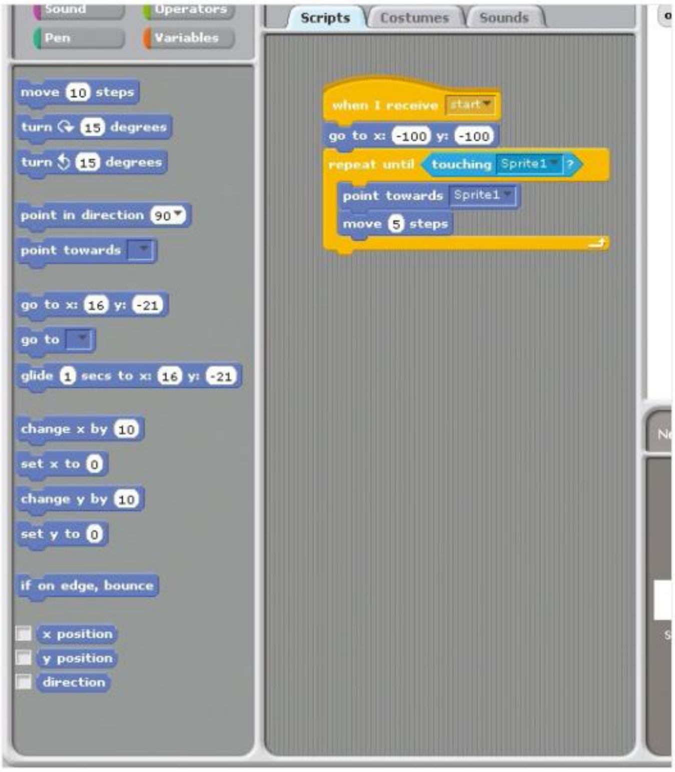
To import a cat, click Choose New Sprite From File above the sprites panel at bottom-right, select Cat 4, then shrink it to an appropriate size, as we did with the mouse. Each sprite has its own set of scripts. To swap between them, click the icon you want in the sprites panel.

12 Move the cat

In the scripts for the new sprite, start a new script with **When I Receive start** (from Control), and add **Go To X:-100 Y:-100** (from Motion). This will move the cat over to the opposite corner of the screen from the mouse. (0,0) is the middle.


Pen
Variables
touching ?
touching color ?
color is touching ?
ask What's your name? and wait
answer
mouse x
mouse y
mouse down?
key space pressed?
distance to
reset timer
timer
x position of Sprite1
loudness
loud?
slider sensor value
sensor button pressed ?
Scripts
Costumes
Sounds

```
when I receive start
go to x: -100 y: -100
repeat until touching Sprite1 ?
```


Sound
Operators
Pen
Variables
move 10 steps
turn 15 degrees
turn 15 degrees
point in direction 90
point towards
go to x: 16 y: -21
go to
glide 1 secs to x: 16 y: -21
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
x position
y position
direction
Scripts
Costumes
Sounds

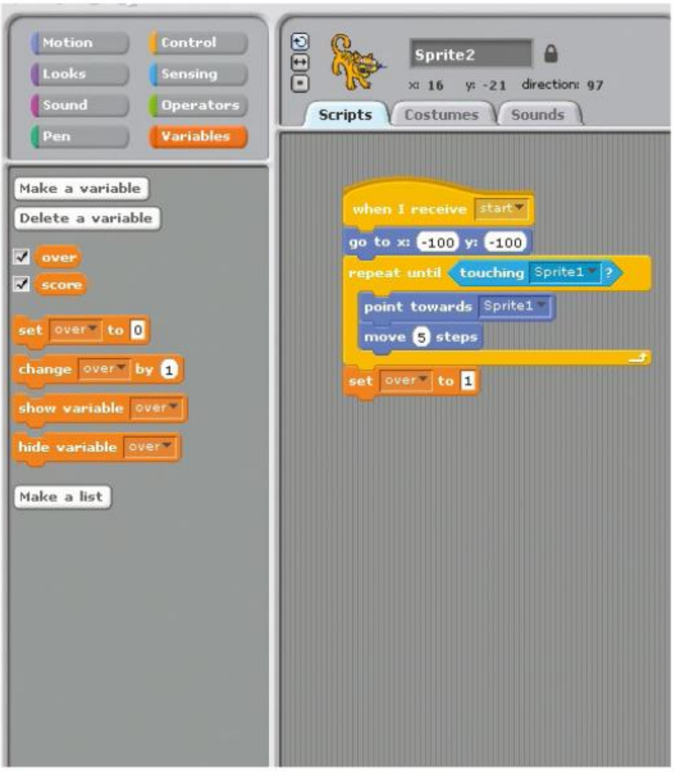
```
when I receive start
go to x: -100 y: -100
repeat until touching Sprite1 ?
    point towards Sprite1
    move 5 steps
```

13 Give the cat a loop

Like the mouse, the cat also needs a loop to keep things going. Add **Repeat Until** (from Control), and then in the blank space add **Touching Sprite 1** (from Sensing). This will keep running until the cat (sprite 2) catches the mouse (sprite 1).

14 Set the difficulty

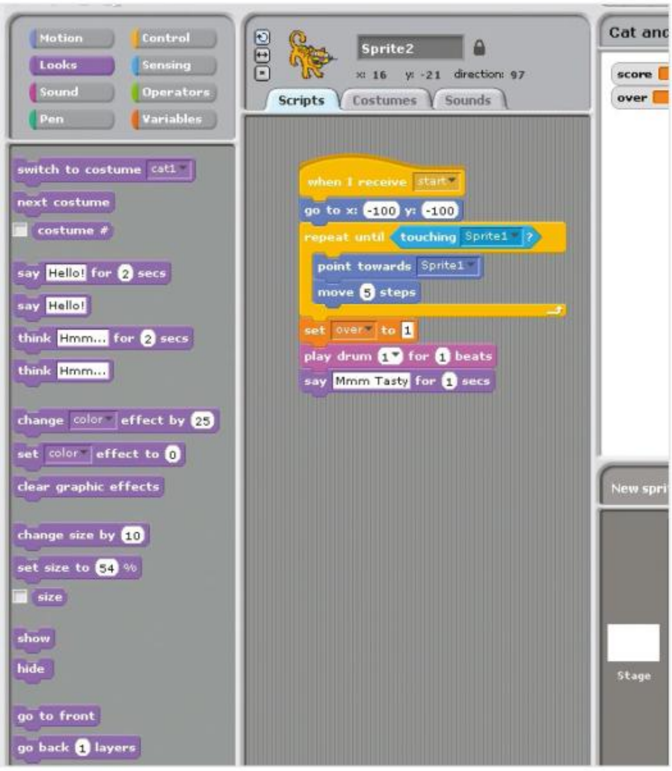
Inside the **Repeat Until** block, add **Point Towards Sprite 1** (from Motion) and **Move 4 Steps** (also from Motion). The amount the cat and mouse move in each loop affects the difficulty of the game. We found that 4 and 7, respectively, worked well.


Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
Make a variable
Delete a variable
☑ over
☑ score
set over to 0
change over by 1
show variable over
hide variable over
Make a list
Sprite2
x: 16 y: -21 direction: 97
Scripts
Costumes
Sounds

```
when I receive start
go to x: -100 y: -100
repeat until touching Sprite1 ?
    point towards Sprite1
    move 5 steps
set over to 1
```

15 Finish the loop

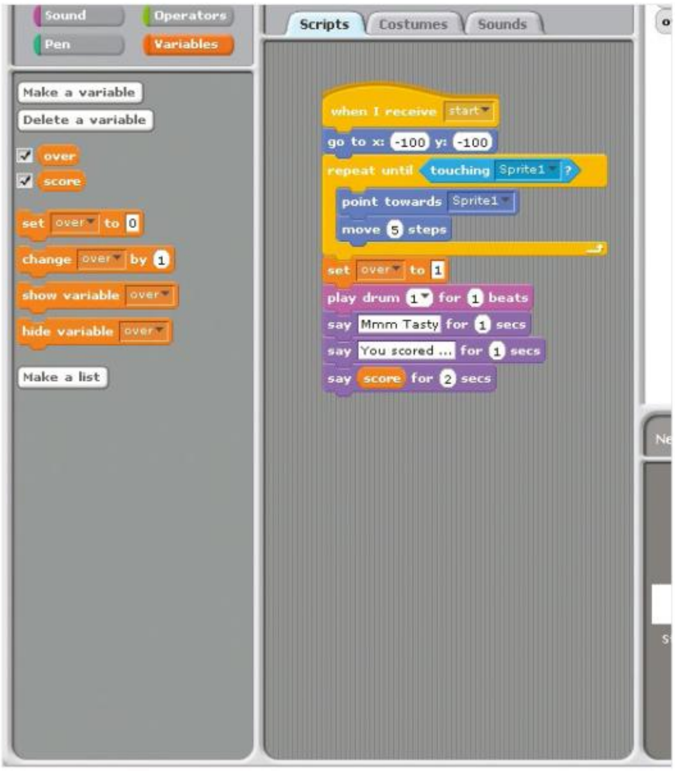
The loop will finish when the cat has caught the mouse – the game is over, so we need to stop the script on Sprite 1. We do this by adding **Set over To 1** (from Variables) underneath the **Repeat Until** block. This will cause Sprite 1's main loop to finish.


Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
switch to costume cat1
next costume
costume #
say Hello! for 2 secs
say Hello!
think Hmm... for 2 secs
think Hmm...
change color effect by 25
set color effect to 0
clear graphic effects
change size by 10
set size to 54 %
size
show
hide
go to front
go back 1 layers
Sprite2
x: 16 y: -21 direction: 97
Scripts
Costumes
Sounds
Cat and
score
over
New spri
Stage

```
when I receive start
go to x: -100 y: -100
repeat until touching Sprite1 ?
    point towards Sprite1
    move 5 steps
set over to 1
play drum 1 for 1 beats
say Mmm Tasty for 1 secs
```

16 Tell the player the game is over

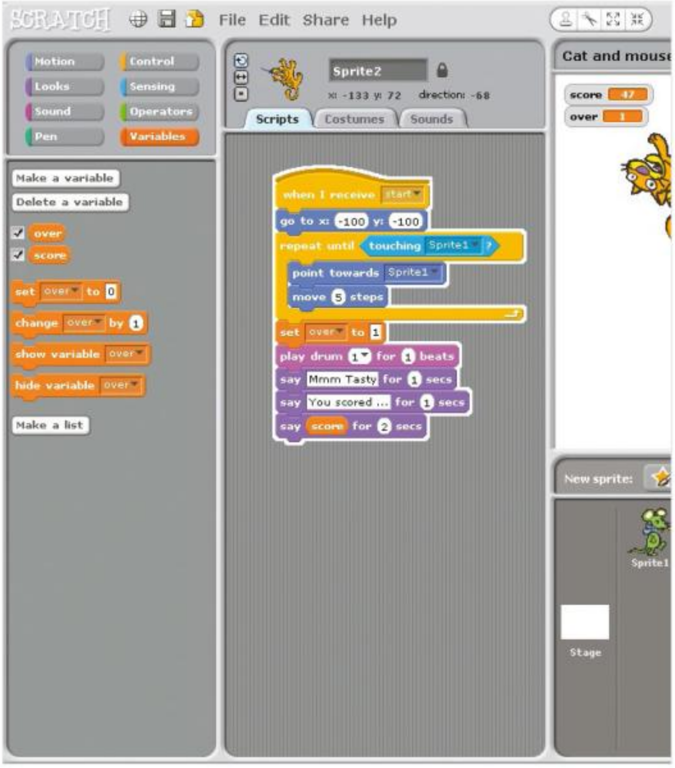
We now want to let the player know that the game is over. We will do this in two ways: with audio and by displaying a message on screen. Add **Play Drum 1 for 1 Beats** (from Sound), then **Say Mmmm Tasty For 1 Secs** (from Looks).


Sound
Operators
Pen
Variables
Make a variable
Delete a variable
☑ over
☑ score
set over to 0
change over by 1
show variable over
hide variable over
Make a list
Scripts
Costumes
Sounds

```
when I receive start
go to x: -100 y: -100
repeat until touching Sprite1 ?
    point towards Sprite1
    move 5 steps
set over to 1
play drum 1 for 1 beats
say Mmm Tasty for 1 secs
say You scored ... for 1 secs
say score for 2 secs
```

17 Display a score

Finally, we can let the player know their score. We increased the variable **score** by one every loop, so this will have continued to go up. Add **Say You Scored ... For 1 Secs**, then drag another **Say ... for 1 Secs** block and then drag **score** (from Variables) into the blank field.


SCRATCH
File
Edit
Share
Help
Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
Make a variable
Delete a variable
☑ over
☑ score
set over to 0
change over by 1
show variable over
hide variable over
Make a list
Sprite2
x: -133 y: 72 direction: -68
Scripts
Costumes
Sounds
Cat and mouse
score 47
over 1
New sprite:
Sprite1
Stage

```
when I receive start
go to x: -100 y: -100
repeat until touching Sprite1 ?
    point towards Sprite1
    move 5 steps
set over to 1
play drum 1 for 1 beats
say Mmm Tasty for 1 secs
say You scored ... for 1 secs
say score for 2 secs
```

18 Play your game!

Click the disk icon in the top bar to save, then press **r** and play! You can use the up and down arrows to move the mouse around. You can make it easier or more difficult by changing the size of the sprites and the amount they move each loop. Good luck and happy gaming! ■

Take your first steps coding with Python

Robert Jones helps you begin learning the basics of the world's most popular programming language on your Raspberry Pi.

There's a multitude of programming languages, but if you're going to learn just one, then Python is the one. Python is used by students around the world to learn coding skills and by big companies such as banks to run mission critical software.

There are many reasons why Python is such a favourite. It's easy to learn but at the same time incredibly powerful – it's like getting a learner bike, complete with training wheels, that's good enough to enter the Tour De France. It is easy to read Python code, and it goes light on syntax. Semi-colons and curly brackets – { and } – litter languages like C and Java, while Python just uses indentation and line spacing to organise code.

GPIO in Python

The great thing about the Raspberry Pi is the GPIO pins you can access on the device. These enable you to hook up hardware, connect circuits and build all kinds of projects. It's the interface between your software and hardware. You can control the GPIO Pins using Python (along with many other languages). Support for GPIO is built into the Raspberry Pi version of Python, so you don't need to set anything up.

There is also a more friendly interface to GPIO called GPIO Zero. This provides a lot of component interface for devices such as LEDs and sensors. You can learn more about GPIO Zero at its main website, **https://gpiozero.readthedocs.io/en/v1.3.1/**.

Because the Raspberry Pi is all about learning to code, it's no surprise to find Python already in the system. Click on Menu > Programming, and you'll find two versions of Python available: Python 2 (IDLE) and Python 3 (IDLE).

There are differences in the way Python 2 and Python 3 code works. Even the humble "print" statement works differently.

Starting Python

Most tutorials and resources are created with Python 2 in mind, and most university courses teach Python 2. So it's best to start with Python 2 and move to Python 3 when you have a few lines of code under your belt. So choose Python 2 (IDLE) from the menu bar.

A window opens displaying "Python 2.7.9 Shell" and three chevrons: ">>>". This window is known as interactive mode, and you can enter Python commands straight into it. It's good luck to christen all new programming languages by printing the traditional message "Hello, World!" to the screen. Input the following command:

```
print "Hello, World!"
```

The phrase "Hello, World" is outputted as a string and appears below your command. You then get another command prompt.

And so you can start programming, entering commands as input and getting direct output. This is known as "interactive mode". We'll come to “script mode” in a minute, but first we’ll look at some maths.

Python Math

Python works well as a calculator. To demonstrate this, type in the following:

```
10 + 20 - 5
```

You’ll get 25 as output. Notice that you didn’t need to enter “print” at the start. It outputs the answer as an integer. You can put “print” before it if you like; you’ll get the same answer.

Now try the following. But before you enter it, try to guess the answer…

```
10 + 10 * 2
```

You’ll be forgiven for thinking 40. But the answer comes back as 30. Why? The reason is called “operator precedence”. Python, like all computer programs, doesn’t work from left to right. Instead, it considers the “*” and “/” operators to be more important than the “+” and “-”. So our 10 + 10 * 2 program goes like this:

```
10 * 2
10 + 20
30
```

The power symbol, which is “**” in Python, as in 2 ** 4 (two to the power of four) is even more important still. If you want to get around all this, you place calculations you want to be performed first in parentheses, like this:

```
(10 + 10) * 2
```

Now you’ll get the answer you want: 40.

Script mode

You can use interactive mode as a calculator, and experiment with basic commands. But to string commands together (to create that thing we call a “program”), you’ll need to create a new file and use Script mode.

Go to File > New File and a new window will open called “Untitled”. Now go to File > Save and name the new file **hello.py**.

Now we’re going to add a one-line program that says hello to you by name. Add these lines:

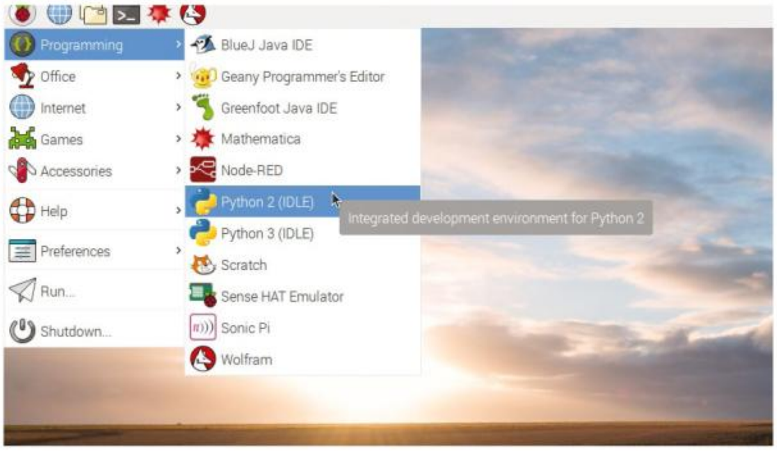
```
name = "Robert"
print "Hello " + name
```

Don’t forget to include to space after “Hello” or it’ll say “HelloRobert”. Go to File > Save and press F5 to run the program. The Shell window returns, and displays “Hello Robert”.

The element “name” is called a “variable”. This is because the thing it contains can vary. In this instance, it includes the name “Robert”. Variables that contain text are known as “strings” as opposed to numbers (which are often referred to as “integers”).

The joy of variables is that you can use them again and again, and quickly change them. Create a new file and name it **movie_script.py**. Enter this code to recreate a famous movie script:

```
person = "Dave"
computer = "HAL"

print "Hello, " + computer + ".Do you read me, " +
computer + "."
print  "Affirmative, " + person + ". I read you."
print "Open the pod bay doors, " + computer
print  "I'm sorry, " + person + ". I’m afraid I can’t
do that."
```



There are two versions of Python installed in Raspbian. Python 2 is the one you should start learning.

Press F5 and you’ll see the script appear in Python Shell. Now change the strings in the variables **person** and **computer**, so the person is called “Bob” and the computer is called “Pi”. Press F5 again and you’ll see the new script.

Variable types

Remember how we said that variables could be strings or integers (and other types)? Python gets a bit confused if you try to mix the two together. This is particularly challenging when combining numbers with strings. First enter this text:

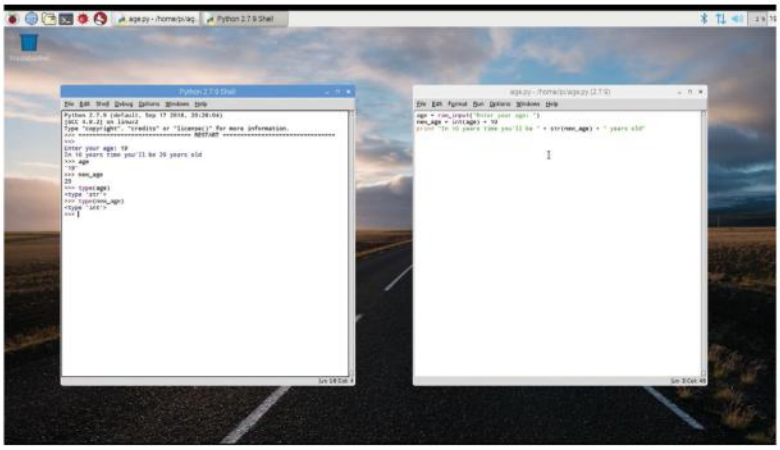
```
answer = 42
print "The answer to the meaning of life, the
universe, and everything is " + answer
```

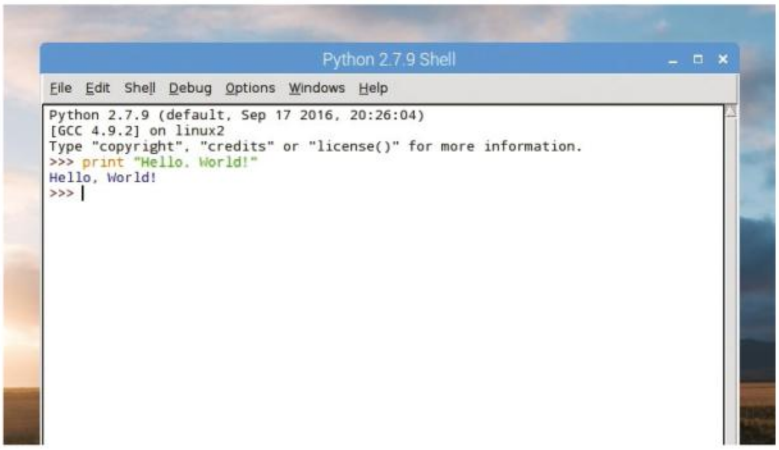
Press F5 and you’ll get an error message: “TypeError: cannot concatenate ‘str’ and ‘int’ objects”. This error arises because answer is an integer, and the print statement needs a string. Python cannot work with them.

You could change **answer** to a string by putting the 42 in quotes: `answer = "42"`. But then you won’t be able to do any maths with it. It’s no longer an integer, and you can’t perform maths with strings. A lot of the time you’ll want to do some maths and then print out the result with a message. To do that you need to convert the integer into a string, using a function called `str()`. You can spot functions because they have two brackets (parentheses) after them.

»

Python can be run in Interactive Mode and in Script Mode. In Interactive Mode you can enter commands and change program variables from the command line.





```
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> print "Hello, World!"
Hello, World!
>>> |
```

Traditionally the first program you run in any new programming language displays “Hello World” on the screen.

The `str()` function takes a number and spits it back out as a string. You put the number inside the brackets, like this: `str(42)`. The process is called “typecasting” (yes, really) and it turns one type into another. If you want to typecast a string into a number, you use the `int()` function. Place the string inside, as in `int("42")`, and you’ll get 42 back as an integer.

So let’s try another program:

```
answer = 7 * 6
print "The answer to the meaning of life, the universe, and everything is "  + int(answer)
```

Typecasting is most useful when you’re using `raw_input()` to request a number from the user. Let’s try the following code:

```
age = raw_input("Enter your age: ")
new_age = int(age) + 10
print "In 10 years time you'll be " + str(new_age) + "years old!"
```

First we see a new function, called `raw_input()` with “Enter your age: ” inside the brackets. This function displays the message inside the brackets, then takes the input received from the user and returns it as a string. It’s then stored (as a string) in the **age** variable.

On the second line we typecast the **age** variable from a string into an integer, and then add 10 to it. This number is stored in the **new_age** variable as an integer.

On the third line, we print out the message and convert the **new_age** variable into a string so it can be printed along with our message.

Back to interactive

We’re now going to head back to the Python 2.7.9 Shell to show something powerful about Python. After you’ve run a program in Script mode, you can play around with it in Interactive mode. Run the age program above, and enter 19 as your age. You’ll see the message “In 10 years time you’ll be 29 years old”.

Now enter **age** in interactive mode. It’ll say ‘19’. Next enter **new_age** in interactive mode. It’ll say 29. Notice that ‘19’ has single quote marks around it, and 29 doesn’t. Quotation marks are a visual way to tell that the **age** variable is a string, while **new_age** variable is an integer.

Another way is to use the `type()` function. This function lets you know what type a variable is. You place the variable, as always with functions, inside the brackets.

Enter `type(age)` and it’ll return `<type ‘str’>`

Enter `type(new_age)` and it’ll return `<type ‘int’>`

This is a powerful feature that Python has (and most other programming languages lack). After you’ve run a program, you can mess around with the variables on the command line, checking and changing them to see what happens. This makes it much easier to figure out what’s going on in your program.

What Else (and If)

So far we’ve looked at programs that can ask for information, store it as variables, manipulate the variables and output it back to the user. We’ve come quite a way.

Of course, this isn’t enough. Programs make decisions based on data (in this case, your variables). To make decisions we use three statements: `if`, `else` and `elif` (else if). Let’s see how this works:

```
password = "knockknock"
entry = raw_input(“Enter password”)
if entry == password:
    print "Welcome!"
else:
    print "Wrong password. Access denied."
```

There’re a couple of things here to observe. First, on line three we introduce the “if” statement and immediately follow it with “**==**”. This is completely different from the “**=**” symbol. The single “=” is used to set a variable. If you had put `entry = password` then both would be set to “knockknock”, the same as if you’d put `entry = “knockknock”`. But the “==” symbol is used to check whether both things are the same. Confusing these two symbols is one of the most common errors you’ll make. Check carefully.

The fourth line is indented by four spaces. You can also press Tab to indent, but you should never mix and match. Pick one or the other.

Press F5 to run the program. You’ll be asked to enter a password, and if you enter “knockknock”, you’ll get the welcome message. You now have an interactive program that responds to your input.

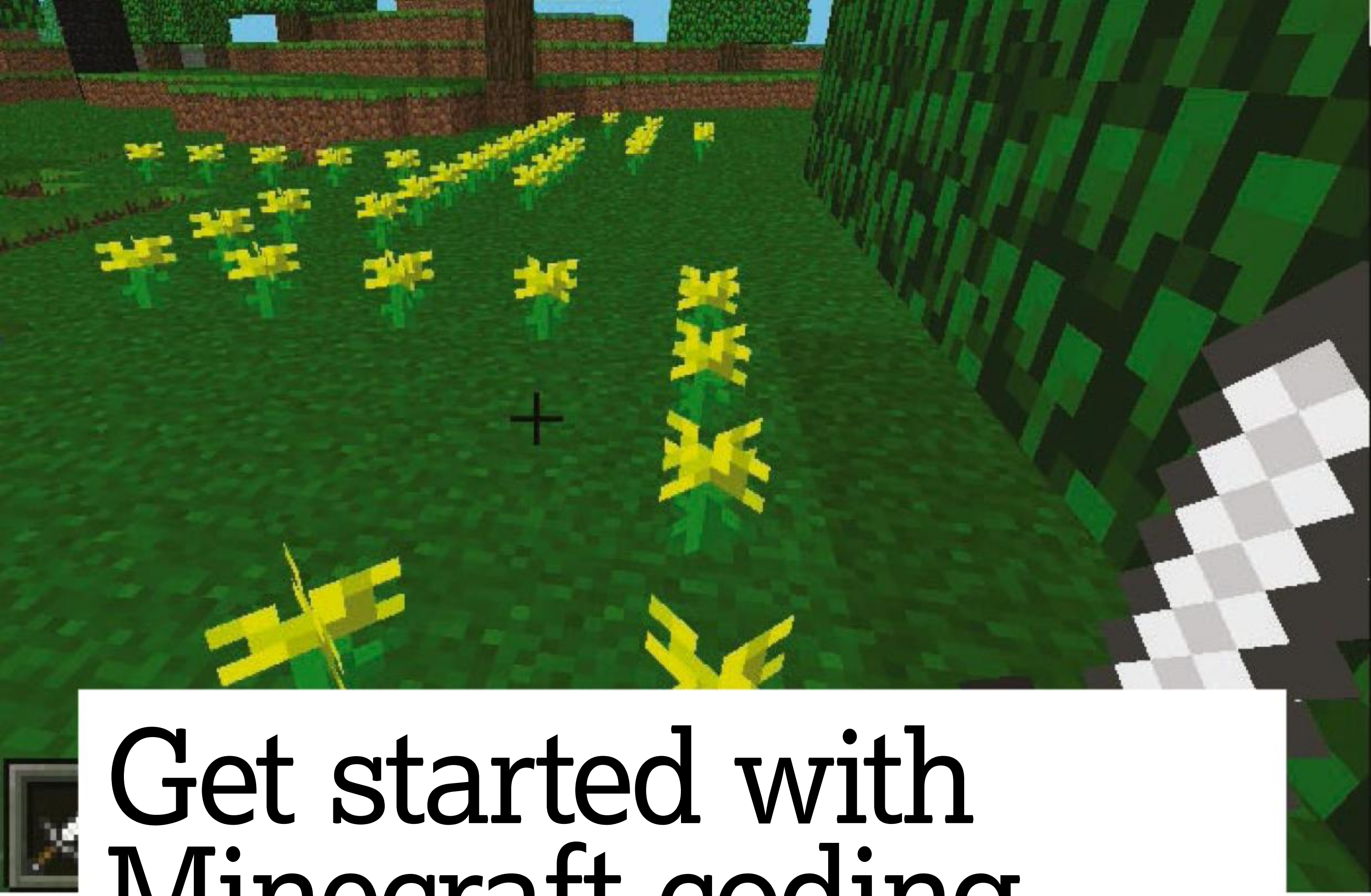
There’s a lot more to learn about Python, but the Raspberry Pi is the ideal environment to learn it. You can experiment to your heart’s content! ■

Python in Terminal

Python isn’t just accessed in the IDLE app. It’s also part of the Raspbian Linux system, and you can run Python programs in Terminal. You can also run Python in Interactive mode (although in this mode you can’t run programs and then access and adjust variables as you would in IDLE).

In Terminal, enter `python` to access Interactive mode. You can perform quick math calculations and run simple scripts.

To run a python program, simply enter `python` and the name of the program (the file ending in .py).



Get started with Minecraft coding

Robert Jones explains how to hack into Minecraft and build your own world with this easy introduction to the Minecraft API

Minecraft is one of the most popular games ever made. In it you control a character, officially called "the player" but often called Steve or Alex. The world of Minecraft is made of blocks. In it, you dig up the blocks and craft them into various things – houses, cars, statues, boats and so on. It's like virtual Lego.

Minecraft for Raspberry Pi is a smaller version of the pocket edition. It may have fewer features than the full version, but it has an ace up its sleeve. You can hack Minecraft on the Raspberry Pi using programming languages such as Python and Java. This means that using just a few lines of code, you can change the Minecraft world around you, adding and removing blocks, changing blocks and moving the player around.

Other players have created countless programs for Minecraft on the Raspberry Pi. These are shared online, and you can import them, and examine them. People have built giant randomly generated mazes, working clocks and even video games you can play inside Minecraft. This makes Minecraft an ideal environment for learning the basics of programming. It's a highly visual world, and you can see the results of your code in a virtual world. And there's plenty of code for you to play with. Above all that, it's fun!

Minecraft is installed by default in the Raspbian Jessie with Pixel operating system. Start it up by going to Menu > Games > Minecraft Pi. Click Start Game > Create new to start playing the game normally. You look around with the mouse, and move with the W,A,S,D keys.

Playing Minecraft normally isn't what we're here for, though. So let's get to know the Minecraft API. Press the Esc key on your keyboard to stop the game (but don't close it down).

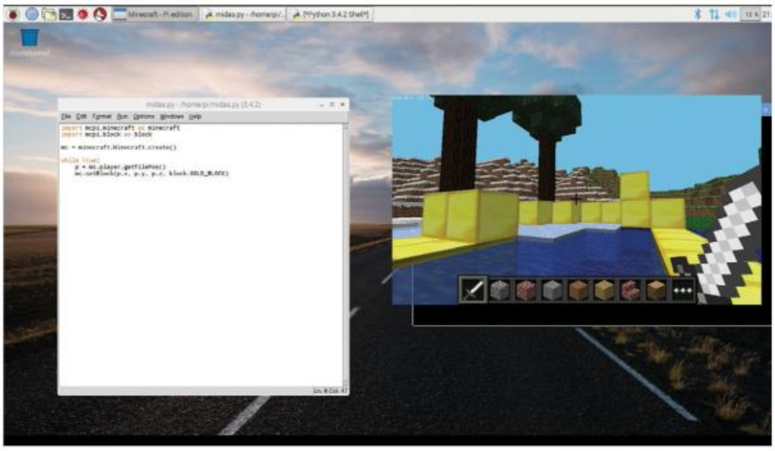
The Minecraft API

The term API stands for Application Programming Interface. APIs are big blocks of code made by other people that you can use. With an API you just need to learn a few commands, and the API will do all the heavy lifting.

In Minecraft, the API works by changing the "server". In this context, this is not a remote computer on which data is stored. Rather, this is a program which runs underneath the game. With the API you can interact with blocks and the player. Here are some of the things you can do:

Get the player's position.
Change (or set) the player's position.
Get the type of block.
Change a block.
Change the camera angle.
Post messages to the player.

We're going to use the Python API for Minecraft. As is tradition, the first program we'll create will send the player the message "Hello World."



Transform regular blocks into solid gold (or any material you choose) using the setBlock method.

So, to get started, return to Minecraft and go to Menu > Programming > Python 3 (IDLE). Use File > New File to open a new window. To save the file go to File > Save, and name it minecraft.py.

Next import the Minecraft API into your Python program by typing this line of code in the file:

```
import mcpi.minecraft as minecraft
```

Now create a connection between your program and Minecraft and call it **mc** (which stands for "Minecraft connection"):

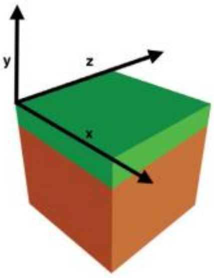
```
mc = minecraft.Minecraft.create()
```

Now use **mc** (the Minecraft connection) to send a command to the game:

```
mc.postToChat("Hello World!")
```

Save the program. Return to the Minecraft game and press Esc again. You'll now be back in the game and will briefly see the message "Hello World!" displayed on the screen. Congratulations, you just took your first step into meddling with the Minecraft world.

Minecraft's unconventional 3D coordinate system.



Move the player

Obviously just having "Hello World" appear on the screen, although it's a nice first step, isn't very exciting or useful. So let's do something a bit more adventurous in our adventure.

In the top-left of the Minecraft screen, you'll see **pos:** and three numbers (initially all three are 0). This represents the position, and the three numbers are x, y and z coordinates. This time, press Alt + Tab to switch between Minecraft and IDLE. With this method, you can make adjustments while Minecraft is still running.

Unlike in normal coordinate systems, the x and z variables are for the position of the player on the ground. Move around, and you'll see them change. The y variable is for the height. Each number represents one block, so if you walk straight forward (without turning), you'll see the z variable increase by one for every block you walk over.

Let's drop the player in mid-air. Delete the line of code with **mc.postToChat("Hello World!")** but leave the first two lines of the code (the ones that import minecraft and create the mc connection). Now enter this line of code:

```
mc.player.setPos(0,100,0)
```

Save the program and press F5 to run it. The player teleports to mid-air and falls back down to the ground. Try some other coordinates to learn how they affect your hapless player!

Block party

Now that you've learned to print messages on the screen and move the player around, it's time to get down to business. And in Minecraft that means blocks. With the Minecraft API, you can turn any block into any other type of block. Turn sand into air, or flowers into iron ore.

We're going to turn the player into King Midas, turning blocks into gold as he walks around. Open IDLE and choose File > New (and save the file as midas.py). For this we need to import minecraft and block from the mcpi file:

```
import mcpi.minecraft as minecraft
import mcpi.block as block
mc = minecraft.Minecraft.create()
```

To get the blocks to respond to the player, we will need to create a variable that stores his position. Traditionally this is stored as a variable called "p":

```
p = mc.player.getTilePos()
```

Now we're going to change the tile beneath the player into gold using the setBlock method.

```
mc.setBlock(p.x, p.y-1, p.z, block.GOLD_BLOCK)
```

We use **p.x** and **p.z** to get the player's horizontal position but **p.y-1** to use the block beneath the player – remember that y is the vertical position.

Press F5 to run the program and look down. You'll see that the block has turned to gold. But just one gold block isn't enough. So let's get the program to loop forever by indenting the two lines inside a "while true" statement, so it looks like this:

```
while True:
    p = mc.player.getTilePos()
    mc.setBlock(p.x, p.y-1, p.z, block.GOLD_BLOCK)
```

The "while True:" code creates an infinite loop, so this program will keep running until you stop it. Press F5 and now, as you run around, a trail of gold blocks appears behind you. They even appear in the air. Change the GOLD_BLOCK to ICE to leave a trail of ice behind instead. Check out the *Blocks*

STROLL AROUND NEW YORK

You can use the Minecraft API to render 3D models as object files (OBJ) into Minecraft worlds. Martin O'Hanlon has a great example where he's created New York by downloading a Google SketchUp model of Manhattan and using OBJexporter (see **http://bit.ly/OBJexporter**) to export the model. This is then rendered inside Minecraft on the Raspberry Pi.

Open Terminal and enter the following code:

```
cd ~
git clone https://github.com/martinohanlon/minecraft-renderObj.git
cd minecraft-renderObj
python minecraft-renderObj.py
```

Be warned that this can take a long time (hours) to render. However, you can walk around as it's being built. If you find yourself stuck inside a tower block, use IDLE to move the player:

```
import mcpi.minecraft as minecraft
mc = minecraft.Minecraft.create()
mc.player.setPos(0,100,0)
```

You can read more about this and see some images on Martin's website, stuffaboutcode.com, at **http://bit.ly/Minecraft-NY** *in the Minecraft API* box below for a full list of available materials.

It is also possible to change blocks when the player hits them with his sword (by right-clicking the mouse). You do this by creating a **hit** variable, which detects an event. Remove the three lines of “while true” code from the program above, and enter the following instead:

```
while True:
    for hit in mc.events.pollBlockHits():
        mc.setBlock(hit.pos.x, hit.pos.y, hit.pos.z, block.GOLD_BLOCK)
```

Press F5 to run the program. Now as you hit blocks, they’ll turn to gold.

You can create lots of blocks at once in Minecraft. You do this by passing in two sets of positions to **setBlocks**. The Minecraft API fills in the blanks.

We’re going to create the black obelisk from the movie *2001: A Space Odyssey*. Get rid of all the code in IDLE and enter this:

```
import mcpi.minecraft as minecraft
import mcpi.block as block
mc = minecraft.Minecraft.create()
p = mc.player.getTilePos()
mc.setBlocks(p.x + 1, p.y, p.z + 1, p.x + 10, p.y + 5, p.z + 10, block.OBSIDIAN)
```

Save, press F5, and in the game the black tower of blocks will appear just next to you.

Finding blocks

You can detect a block type underneath the player using the **getBlock()** method. This method returns a number, which you can check against the API list using an **if** statement.

We’re going to create a program that turns grass into flowers as the player walks around. We’ll do this by checking what is beneath the player using **getBlock(p.x, p.y-1, p.z)** and storing this in a variable called **below**.

Then we check to see if the **below** variable matches **grass** (which has an ID number of 2). If we get a match, we’ll use **setBlock()** to change the block from GRASS to FLOWER_YELLOW. (The Minecraft API is very forgiving: you can use ID numbers or names for materials, or mix the two.)

Enter this code in IDLE:

```
import mcpi.minecraft as minecraft
import mcpi.block as block
mc = minecraft.Minecraft.create()

while True:
    p = mc.player.getTilePos()
    below = mc.getBlock(p.x, p.y - 1, p.z)
    if below == 2:
        mc.setBlock(p.x, p.y, p.z, block.FLOWER_YELLOW)
```

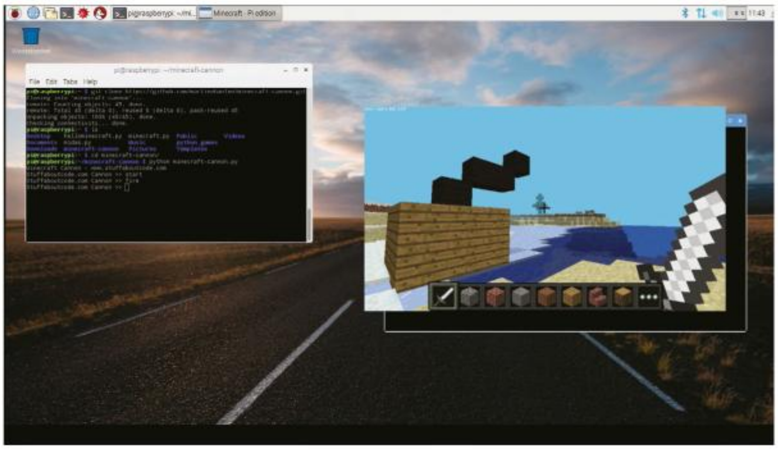
Press F5 to run the program. Now as the player runs around, the blocks of grass where he stands will turn into flowers.

Getting other programs

Now that you’ve got the basics of the Minecraft API down and created your first program, you’ll want to look at something a little more complex.

Fortunately, there are lots of examples of Minecraft API coding on the Internet. You can download these programs, and play around with them in your own copy of Minecraft. Not only are many of the programs fun to play in their own

»



Import code created by other users to add powerful new features to Minecraft, such as this working cannon.

Pi facts

If you connect multiple Raspberry Pis to a local network, players can join the same game and play together. They can then see each other in the Minecraft world.

BLOCKS IN THE MINECRAFT API

Here are the blocks you can access in the API, along with their ID codes.

AIR	**0**
STONE	**1**
GRASS	**2**
DIRT	**3**
COBBLESTONE	**4**
WOOD_PLANKS	**5**
SAPLING	**6**
BEDROCK	**7**
WATER_FLOWING	**8**
WATER	**8**
WATER_STATIONARY	**9**
LAVA_FLOWING	**10**
LAVA	**10**
LAVA_STATIONARY	**11**
SAND	**12**
GRAVEL	**13**
GOLD_ORE	**14**
IRON_ORE	**15**
COAL_ORE	**16**
WOOD	**17**
LEAVES	**18**
GLASS	**20**
LAPIS_LAZULI_ORE	**21**
LAPIS_LAZULI_BLOCK	**22**
SANDSTONE	**24**
BED	**26**
COBWEB	**30**
GRASS_TALL	**31**
WOOL	**35**
FLOWER_YELLOW	**37**
FLOWER_CYAN	**38**
MUSHROOM_BROWN	**39**
MUSHROOM_RED	**40**
GOLD_BLOCK	**41**
IRON_BLOCK	**42**
STONE_SLAB_DOUBLE	**43**
STONE_SLAB	**44**
BRICK_BLOCK	**45**
TNT	**46**
BOOKSHELF	**47**
MOSS_STONE	**48**
OBSIDIAN	**49**
TORCH	**50**
FIRE	**51**
STAIRS_WOOD	**53**
CHEST	**54**
DIAMOND_ORE	**56**
DIAMOND_BLOCK	**57**
CRAFTING_TABLE	**58**
FARMLAND	**60**
FURNACE_INACTIVE	**61**
FURNACE_ACTIVE	**62**
DOOR_WOOD	**64**
LADDER	**65**
STAIRS_COBBLESTONE	**67**
DOOR_IRON	**71**
REDSTONE_ORE	**73**
SNOW	**78**
ICE	**79**
SNOW_BLOCK	**80**
CACTUS	**81**
CLAY	**82**
SUGAR_CANE	**83**
FENCE	**85**
GLOWSTONE_BLOCK	**89**
BEDROCK_INVISIBLE	**95**
STONE_BRICK	**98**
GLASS_PANE	**102**
MELON	**103**
FENCE_GATE	**107**
GLOWING_OBSIDIAN	**246**
NETHER_REACTOR_CORE	**247**

Minecraft commands

We use several Minecraft API commands in this project. There are fewer commands than you imagine, but once you learn them, you can create some incredibly powerful programs. Here is the Minecraft API:

`postToChat(message)` Communicate in the game.

`getBlock(x, y, z)` Get a block type for a specific position.

`setBlock(x, y, z, blockType, blockData)` Change a block to a certain blockType.

`setBlocks(x1 , y1 , z1, blockType)` Create a block.

`setBlocks(x1 , y1 , z1 , x2, y2, z2, blockType)` Set lots of blocks all at the same time by providing two sets of co-ordinates.

`player.getPos()` Get the current position of the player.

`player.setPos(x, y, z)` Set the player's position.

`player.getTilePos()` Get the position of the tile the player is currently on.

For more, see **www.stuffaboutcode.com/p/minecraft-api-reference.html**. With these commands, you can alter Minecraft in any way you want!

Pi bites

You can find the Minecraft API in Terminal using **cd /opt/minecraft-pi/api**. Type the command **cat /opt/minecraft-pi/api/spec/** to view the official specifications for the project.

right, but you can also examine the code and learn how it works.

We're going to use Git to install the software. Git is installed by default in Raspbian Jessie with Pixel, but if you don't have it, you can install it using:

```
sudo apt-get install git-core
```

A good place to start is minecraft-clock. This code draws a large circle, then uses trigonometry to find where the hands should be drawn.

Make sure that Minecraft is running a game. Open a Terminal window and enter this code:

```
cd ~
git clone https://github.com/martinohanlon/
minecraft-clock.git
cd minecraft-clock
python minecraft-clock.py
```

You will see STARTED in the Terminal and a message displays inside Minecraft welcoming you to the clock. Then, a giant clock appears in the sky. Switch back to Terminal using Alt + Tab and press Ctrl + C to quit the program. The clock will stop running inside Minecraft.

Building a cannon

One of the most complex programs you can run is called Minecraft Cannon by Martin O'Hanlon. You can learn more about the code on the "Stuff About Code" website (**stuffaboutcode.com/2013/04/raspberry-pi-minecraft-cannon.html**. To download Minecraft Cannon, open a Terminal window and enter the following, then hit Return:

```
cd ~
clone https://github.com/martinohanlon/minecraft-
cannon.git
```

Enter **ls** and you'll see the minecraft-cannon directory. Navigate to it and run the Python code contained inside to run the program (make sure you're running Minecraft first).

```
cd minecraft-cannon
python minecraft-cannon.py
```

Terminal will display **Stuffaboutcode.com Cannon >>**. You can use this mode to enter commands to control the cannon. To create a new cannon and fire it, for example, you simply use:

```
start
fire
```

Here are all the commands you can use in Minecraft Canon.

start – create (start-up) the cannon

rotate [0-360 degrees] – rotate the cannon

tilt [0-90 degrees] – tilt the cannon upwards

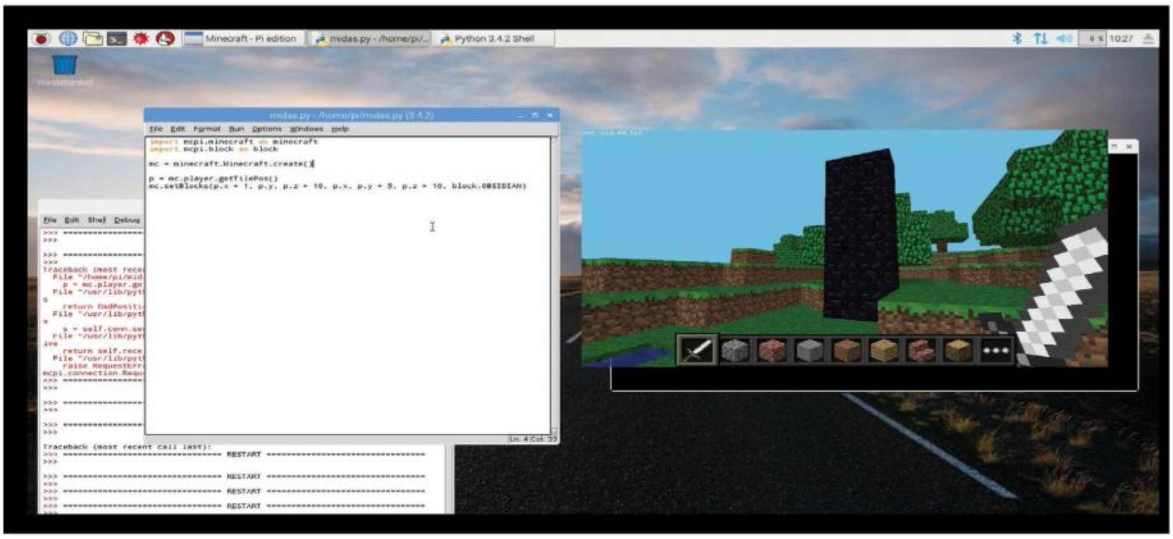
fire – what it says!

exit – exit the program and clear the cannon.

You can investigate the code in Python. Open IDLE from within Minecraft by going to Menu > Programming > Python 3 (IDLE). Now choose File > Open and open the minecraft-cannon directory. Highlight **minecraft-cannon.py** and choose Open to view the code in IDLE.

Here you can see how all the functions have been programmed. It uses the **block** module as well as **time** (to create delays), **math** (to calculate the trajectory) and **cmd** (to interact with the command line).

Minecraft is a tremendously creative game, and with Minecraft API it's possible to create amazing programs to take it even further. This fun gaming connection makes Minecraft API a great way to learn to code, and a good start to learning complex programming languages such as Python (and even Java). As you become more advanced, it's even possible to hook up Minecraft to the Raspberry Pi hardware and blend the real world with Minecraft's imaginary block world. Have fun! ■

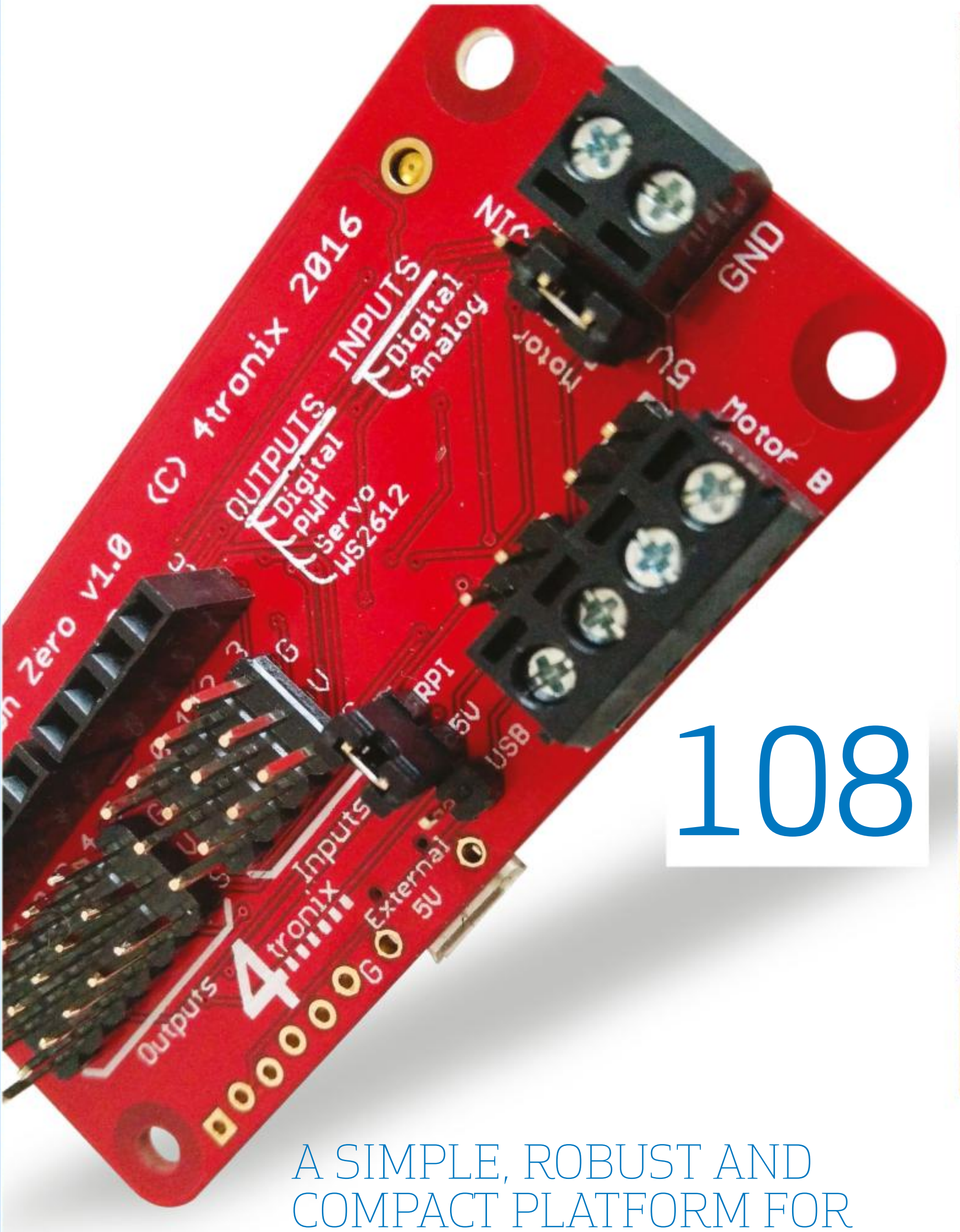


```
import mcpi.minecraft as minecraft
import mcpi.block as block

mc = minecraft.Minecraft.create()

p = mc.player.getTilePos()
mc.setBlocks(p.x + 1, p.y, p.z + 10, p.x, p.y + 5, p.z + 10, block.OBSIDIAN)
```

❯ Create multiple blocks at once to create shapes and structures.

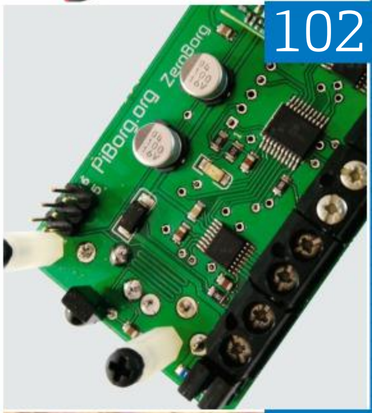
Pi REVIEWS

Tested and rated: the latest add-ons for your Pi



108

100



102

104

A SIMPLE, ROBUST AND COMPACT PLATFORM FOR MANY DIFFERENT PROJECT TYPES, THIS WILL SOON BE YOUR GO-TO BOARD

100 FUZE Workstation

102 PiBorg ZeroBorg

104 Drum HAT

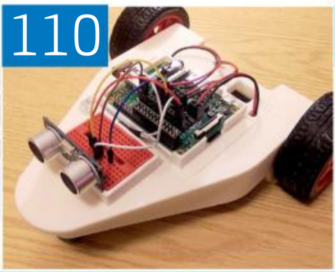
106 Explorer HAT Pro

108 Picon Zero

110 CamJam EduKit 3

112 Picade by Pimoroni

106



110

112

FUZE

Turn your Raspberry Pi into an electronics workstation with the FUZE Computer. **Robert Jones** reviews a great kit for classrooms

In brief

The FUZE is an electronics workstation kit for the Raspberry Pi (which sits inside it). It looks more like a regular computer but has an IO board for connecting electronics components.

The Raspberry Pi is ideal for learning computing and hardware projects. The only problem is that the Raspberry Pi doesn't come complete with a keyboard, mouse or any electronics components. So if you want to use a Raspberry Pi to teach young students electronics projects, you need to add quite a few things to get started.

The FUZE Workstation is built around a Raspberry Pi but looks reminiscent of a home computer from the 1980s. In fact, there's a model with a retro colour scheme designed to look just like the classic BBC Micro (which originally inspired it). In many ways, it's the complete computer setup that many people want.

There's a keyboard on the front and a mouse attached to one of the four USB ports on the rear, where there's also an HDMI connection to connect the FUZE to a television or monitor.

> THE IO BOARD, LIKE A BREAKOUT CABLE, ALLOWS YOU TO CONNECT COMPONENTS SAFELY

The real advantage is the FUZE IO Board that attaches to the GPIO pins of the Raspberry Pi. Like a breakout cable, this allows you to safely connect electrical components to the FUZE without worrying about blowing the Raspberry Pi board.

It also offers additional features, such as analogue input and output. The FUZE Workstation comes with a box of components, including LED lights (red, green and blue), jumper leads, buttons, light sensors, seven segment LEDs and mixed resistors.

FUZE also supplies Project Workbook and Worksheet guides for users to work with these components.

Teachers who want to take things further can use the FUZE with the RC robotic arm from Maplin's. This comes in kit form but is easy to build. It attaches to the FUZE Workstation via USB and software is used to control it.

The only hiccup is that FUZE has put all of its eggs in a Basic programming language basket. The FUZE Workstation runs a stock version of Raspbian and can be programmed in any language. But most of the worksheets and projects use a version of Basic (based on RTB: Return To Basic).

Basic may have retro appeal to teachers remembering days spent entering code line by line on a BBC Micro. But it's been far superseded by entry-level languages like Scratch and Python. We can't see any good

Features at a glance

Case
The FUZE case contains a keyboard and external USB ports that make your Raspberry Pi look more like a regular computer.

Fuze IO Board
On top of the FUZE is an IO (Input/Output) board, which makes it easy to connect electronics to the FUZE workstation.

FUZE
Teach Kids to Code
MagPi
The Register
FUZE
Includes the Raspberry Pi B+
and FUZE BASIC Project Cards

A range of bundles is available, starting at £70 without a Raspberry Pi.

The RC robotic arm from Maplin (shown here) attaches to the FUZE Worksation via USB.

reason to teach young students Basic when you can get them straight into Scratch.

FUZE says its Basic is a stepping stone between introductory tools like Scratch and real-world programming languages such as Python. If you want to investigate the Basic supplied with FUZE, you can download it for free from the FUZE website.

The FUZE can be purchased with a Pi inside (starting at £100 for the basic bundle), or you can buy the kit without a Pi and assemble it yourself. The top-spec kit for £230 comes with Raspberry Pi 3, Programmer's Reference, Project Workbook, 840-pin breadboard, component kit, cables and robot arm kit.

Even if you opt for DIY, putting the kit together is easy. Slot the Raspberry Pi inside the FUZE kit and wire up the external ports to the USB, power and HDMI ports on the Raspberry Pi board.

The Micro SD slot of the Raspberry Pi sits flush against the outer case of the FUZE box. This approach worked fine with the Raspberry Pi 2 and its push-to-eject mechanism, but the Raspberry Pi 3 requires you to pull out the Micro SD card manually. We found that a pair of tweezers helped.

In all, FUZE is a nicely self-contained case for the Raspberry Pi that turns it into a reliable computer with a built-in IO Board for using to learn electronics. Teachers find FUZE incredibly useful for teaching IT in the classroom. ■

Pi User Verdict

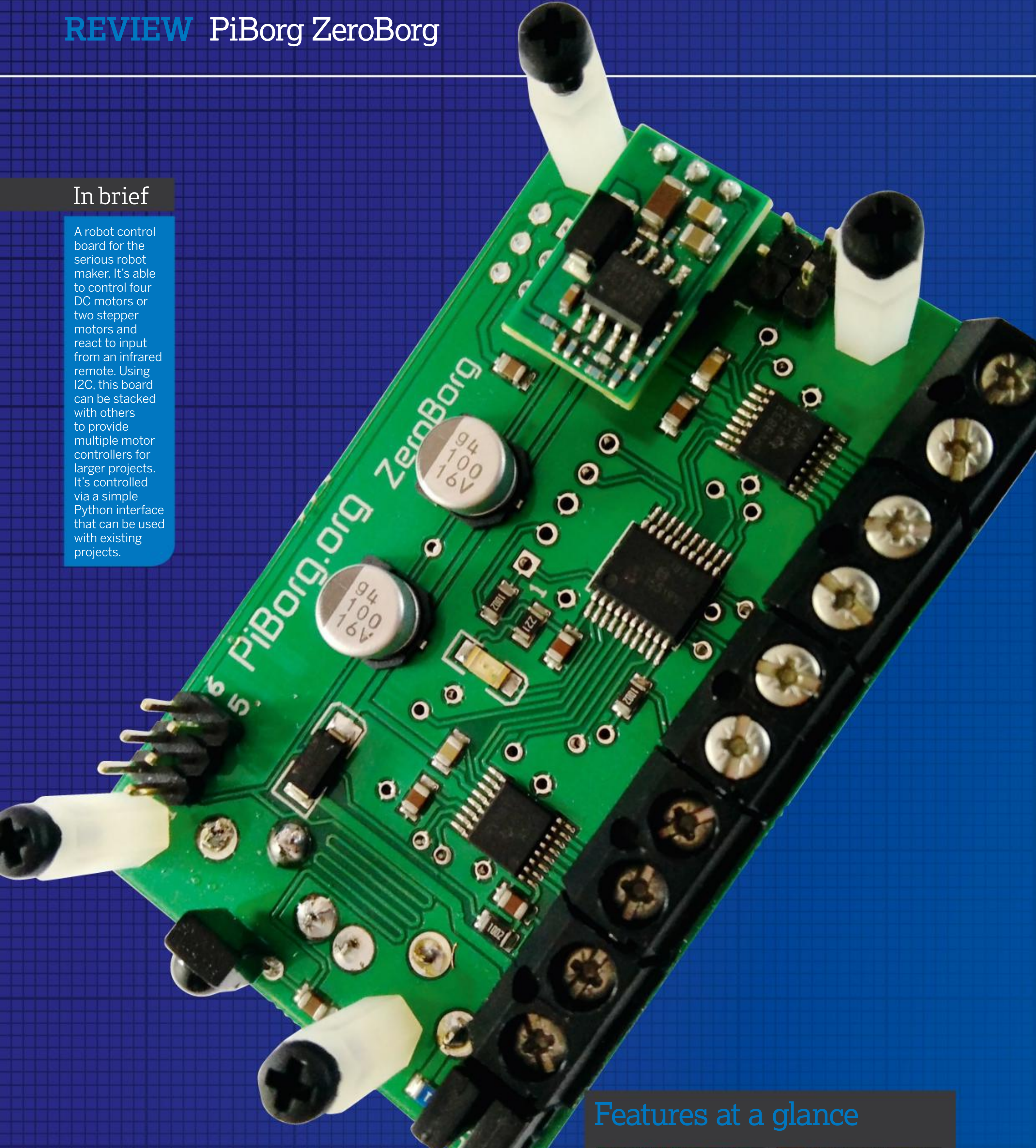
FUZE

Developer: FUZE
Web: www.fuze.co.uk
Price: From £100 including Raspberry Pi 3, or £70 without

Rating 8/10

In brief

A robot control board for the serious robot maker. It's able to control four DC motors or two stepper motors and react to input from an infrared remote. Using I2C, this board can be stacked with others to provide multiple motor controllers for larger projects. It's controlled via a simple Python interface that can be used with existing projects.



Features at a glance

Motor controllers
The ZeroBorg can control four DC motors with individual control of each motor's power.

Infrared input
If you have a spare remote control, you can run the **zbIrRead.py** code to capture each button press.

PiBorg ZeroBorg

Les Pounder prepares to be assimilated by the latest board from the PiBorg roboteers

The PiBorg team is well known for its robotic creations, which include DoodleBorg, a Raspberry Pi powered tank large enough to transport a person. For its latest project the team turned to crowdfunding for a smaller, Pi Zero focused robot control board called ZeroBorg, and it is rather impressive. Measuring 64x40mm, this board is designed for both models of the Raspberry Pi Zero but is compatible with all models of Raspberry Pi as it uses the Inter-Integrated Circuit (I2C) interface, which requires only the first six GPIO pins.

> THE ZEROBORG IS A SERIOUS ROBOT PLATFORM PROVIDING ROCK-SOLID MOTOR CONTROL

The board cradles the Pi Zero from below, with support posts at each corner of the board. Also present on the underside of ZeroBorg is a 9V battery connector, which can power the Pi and the board thanks to a 5V regulator built in to ZeroBorg. The connection between the ZeroBorg and the Pi is made via six General Purpose Input Output (GPIO) pins, and these can be soldered in place or pins can be bent as required. Both of these connections to the GPIO require a Pi Zero with no GPIO header pins soldered. For connection to other Raspberry Pi models, or for a temporary connection, you can use female to female header cables.

Primarily, the ZeroBorg is a motor control board capable of controlling four DC motors simultaneously, thanks to two full H-bridge controllers. (Each one is a DRV8833, which is also used on the 4tronix Picon Zero and the CamJam Robotics kits.) The board can also control two stepper motors, which can be used for slow, precise motion-control applications.

ZeroBorg also comes with analogue inputs for sensors, and the most useful is the infrared receiver, which will work with most remote controls. We tested the receiver with a cheap generic remote, captured the data received by the receiver and used it to control a series of motors attached to the board.

The PiBorg ZeroBorg is a little different from other motor control boards. Given that it uses I2C, a data connection protocol, there can be multiple ZeroBorg controllers in a project, with each giving you control of four motors, so with three units you could drive 12 individual motors.

Controlling the ZeroBorg is handled via a robust Python library, which is installed via an automated script on the PiBorg website. The Python library comes with a series of examples on how to use the ZeroBorg, including how to capture infrared remote signals and use joypads to control your robot. At the time of writing, a Python 3 library is being worked on, and this should be released soon if it's not already available.

The PiBorg ZeroBorg is a serious robot platform providing rock-solid control of motors, both DC and stepper. The ability to add more of these boards to augment a creation is fantastic, as it enables you to combine different motors for varying functionality. The board is easy to use and it took us little time to get going with our infrared-controlled motor project. Because it uses only six GPIO pins, you still have access to the remaining pins, requiring only that you solder a header pin to each GPIO pin you wish to use.

For those just starting out with robotics, something like the Picon Zero or CamJam Robotics kit will be more appropriate because it provides more inputs for sensors. For the serious roboteer, though, this could be the perfect kit to build your next multi-motor monster. ■

Pi User Verdict

PiBorg ZeroBorg

Developer: PiBorg
Web: www.piborg.org/zeroborg
Price: From £18 (as tested £32.50)

Rating 8/10

Drum HAT

Tap your fingers to the beat and turn a Raspberry Pi into a drum machine with this clever HAT. **Robert Jones** channels his inner Fatboy Slim

In brief...

Turn a Raspberry Pi into a drum machine with this cunning piece of HAT hardware. The Drum HAT is an 8-pad drum-kit that fits on top of your Raspberry Pi. It comes packed with samples, and you can quickly turn it into a drum machine. Or you can program it using Python.

The Drum HAT is a crafty piece of kit from Pimoroni. With it, you can transform a Raspberry Pi into a drum machine, tapping out beats on the touch-sensitive surface pads. As this is a HAT (Hardware Attached on Top) kit, setup is simple. All you need to do is attach the Drum HAT to the GPIO pins and load the software from Pimoroni's GitHub (**https://github.com/pimoroni/drum-hat**).

You use Python to assign audio samples to each of the eight pads. Then when you tap the pad with your finger, the audio sample is played. There are two sets of drum samples available in the documentation, each with eight samples (such as clap.wav and ting.wav).

Each pad area comes with a corresponding LED. These LEDs automatically light up when you tap the drum pads, or you can also take control of them manually in code and get them to do what you want.

Setup is so simple that you can get it up and running in a matter of minutes – it just slots on top of the Raspberry Pi. Pimoroni has an installation script on its GitHub page (**https://github.com/pimoroni/drum-hat/blob/master/README.md**). Open Terminal and cut-and-past the curl script and you're ready to go.

USE PYTHON TO ASSIGN AUDIO SAMPLES TO EACH OF THE EIGHT PADS, THEN WHEN YOU TAP A PAD THE SAMPLE IS PLAYED

Alternatively you can use git to clone the software from GitHub (clone https://github.com/pimoroni/drum-hat) and run the Python setup installer. Once it's working, you can quickly scan over the demo scripts and figure out how it works.

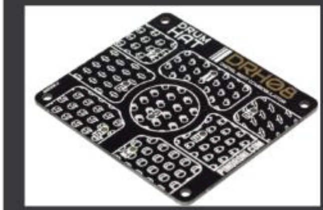
“An 8-pad Raspberry Pi drum-kit that lets your fingers think they're Stubblefield,” claims the blurb. We're not even sure who he is. The built-in samples are limited, but there are hundreds of thousands of beats available online.

Behind the scenes, PyGame is used to connect the audio samples to the touch pads. Inside the Python code is where the real fun is found. You set up the pads using Python event objects. These are an index for each pad (from 1 to 8). You create hits and releases (determining how the pad responds to taps and presses).

If you're really into music on your Raspberry Pi, it's possible to use the Drum HAT alongside Pimoroni's other musical offering, Piano HAT. To do this you'll need an extra piece of kit, the HAT Hack3r, which enables you to connect two pieces of HAT hardware to a single Raspberry Pi.

Serious musicians can also configure the Drum HAT to output MIDI commands via a USB-to-MIDI adaptor. This enables you to connect the Drum HAT and Raspberry Pi kit to a synth setup. Potentially, therefore, you could use it in a professional environment, although we're not sure how reliable it would be in a studio setting. The pads on the Drum

Features at a glance



HAT connection
The Drum HAT fits neatly on top of your Raspberry Pi, turning it into a single box.

Touch pads
Eight touch pads can be individually programmed to respond to finger taps to make a novel controller.

LED lights
Each pad has an LED light that flashes when you tap or can be programmed independently if you prefer.

DRUM HAT
DRH08
DIGITAL BEAT COMBOBULATOR
PIMORONI.COM
F005289

The Drum HAT will sit neatly on top of your Raspberry Pi.

HAT are not pressure-sensitive, either, so it's unlikely to impress a professional drummer or DJ. However, at just £12, it is a fun piece of kit for the Raspberry Pi and does enable budding drummers to have fun with sound and learn PyGame and Python scripting.

We had fun playing around with the Drum HAT. While the projects won't keep you occupied for long, it does have a lively bash and play quality. Music is a great way to get young students interested in hardware and coding in a world that they understand. While Drum HAT doesn't interact directly with Sonic Pi (which is a shame), it could form a good part of a wider course in music and coding. We can envision it working along with MIDI training, Piano HAT and Sonic Pi coding.

It's a great project for the Raspberry Pi on its own, although it is a little simple. That's part of the problem. Other projects have us soldering parts and assembling hardware. By comparison, the Drum HAT's entertainment value is relatively short-lived. But it is good for working through Python and figuring out the audio aspects of the PyGame library. Despite the hype, it's not likely to hit the mark for expert musicians, but it will be good fun for budding musicians looking to learn to code, and useful in education. ■

Pi User Verdict

Drum HAT

Developer: Pimoroni
Web: https://shop.pimoroni.com/products/drum-hat
Price: £12

Rating 8/10

Pimoroni Explorer HAT Pro

Is this the easiest, most convenient way to use your Raspberry Pi for circuit prototyping? **Robert Jones** tips his hat to a cleverly thought out HAT

In brief

A HAT (Hardware Attached on Top) board for prototyping, the Explorer HAT provides four of everything: buffered 5V tolerant inputs, powered 5V outputs, capacitive touch pads, capacitive crocodile clip pads, LEDs and analogue inputs. It also has two H-bridge motor drivers and a convenient mini breadboard on top.

Prototyping circuits is one of the real joys of using a Raspberry Pi. Most of the time this activity involves attaching components to the Raspberry Pi GPIO pins using a breadboard. The Explorer HAT Pro places a breadboard right on top of your Pi and also adds a whole bunch of input and output options that enable you to create prototype circuits right on top of your Pi. It's a bit like having a breadboard and advanced breakout cable all in one place (with a few extra goodies thrown in for good measure).

Since this is HAT hardware, setup is relatively easy. We slotted the Explorer HAT Pro into the GPIO pins. You can also use the included mounts and screws to provide a firmer connection. The mini breadboard is supplied as a separate piece of plastic with an adhesive strip on the rear. You can remove the protective sheet and mount the mini breadboard on top of the HAT hardware, or you can just slot the breadboard on or off as you need it.

Once you've got the Explorer HAT Pro attached, you can start using it (thanks to the EEPROM that automatically connects the hardware to your Pi). To get the most from it, however, you'll also want to use the software that Pimoroni has developed. There's an extensive Python library, which you can learn all about on the Pimoroni GitHub page (**https://github.com/pimoroni/explorer-hat**). Setting up the software side of things is easy, although you'll need to turn on i2c (which can be accessed through the new Raspberry Pi Configuration tool). The Explorer HAT Python library can be imported into your programs and used to control the various components.

> IT'S LIKE HAVING A BREADBOARD AND AN ADVANCED BREAKOUT CABLE ALL IN ONE PLACE, RIGHT ON TOP OF YOUR RASPBERRY PI

Once you have both hardware and software set up, you're ready to party. The setup is ideal for working with 5V powered systems thanks to four buffered 5V tolerant inputs and four powered 5V outputs. These enable you to interface with systems safely, much the same as a breakout cable. There are also two H-bridge motor drivers. These are used to work safely with the high voltages used to drive motor systems and enable you to drive the servo motors used in robotics projects.

There are a heap of extra features on top of the ability to control voltages and motors. Four capacitive touch pads act as buttons. These sit on the side of the Explorer HAT Pro and respond to finger presses. It's easy enough to add buttons to a breadboard, but having the feature built-in makes it that much simpler to prototype circuits that respond to button pushes. You can even use them to turn the Explorer HAT Pro into a drum machine (like a stripped-down version of the Drum HAT).

Alongside the capacitive touch pads are four coloured LED lights: yellow, blue, red and green. These can be controlled

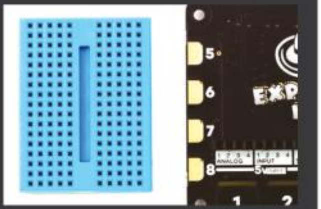
Features at a glance

EXPLORER
HAT Pro
5
6
7
8
1
2
3
4
ANALOG
INPUT
5V
GND
OUTPUT
MOTOR

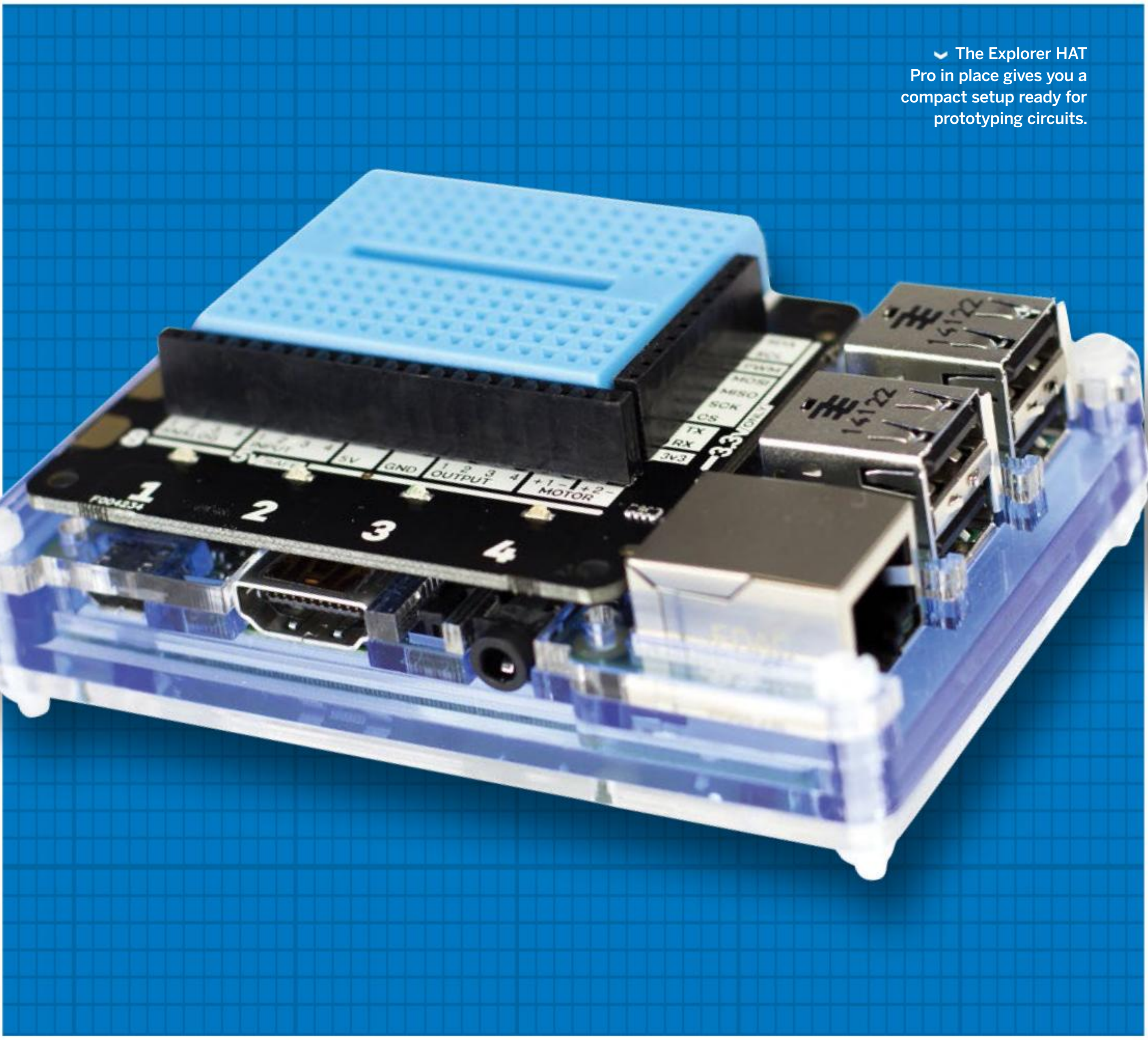
Capacitive touch pads
Four touch pads act as buttons. Responding to finger presses, they can be used to interact with your programs.

Explorer HAT Pro
A multi-talented project board for your Raspberry Pi
Made in Sheffield, UK
http://pimoroni.com

H-Bridge motor drivers
Two H-Bridge drivers enable voltage to be applied across a load in either direction. These are used to control motors.



Mini breadboard
A mini breadboard enables you to prototype circuits and attach components right on top of your Raspberry Pi.


The Explorer HAT Pro in place gives you a compact setup ready for prototyping circuits.

independently for any purpose relating to your circuit.

The board also provides four analogue inputs. The lack of analogue input is limiting on the regular Raspberry Pi board, and the analogue inputs widen your choice of devices. You can use them with potentiometers, photocells, force sensitive resistors, temperature sensors and two-axis joysticks.

On a more regular note, there are four buffered, 5V tolerant inputs. These act just like the GPIO pins on your Raspberry Pi, and you can use them to respond to voltage changes in your circuit (so as to detect motion from a PIR sensor or button pushes and so on).

There are also four crocodile clip pads, of the kind found on the BBC Micro: BIT. While we find these clunkier than the GPIO pins, they are an attractive alternative for many makers. Along with the other features, they add an extra area for you to experiment with.

Pimoroni sells an Explorer HAT Pro Parts Kit (£10) alongside the HAT itself. This kit includes LEDs and resistors plus more interesting components like a Piezo transducer, rotary potentiometers and analogue temperature sensors. All of these parts are readily available, but having the kit will help you work through the features on offer from the Explorer HAT Pro.

The Explorer HAT Pro is a great little prototyping add-on for the Raspberry Pi. It extends the functionality of the GPIO pins in ways that are fun and functional. More importantly, it's a neat, integrated little system that conveniently bundles prototyping into the same space as the Pi board itself. ■

Pi User Verdict

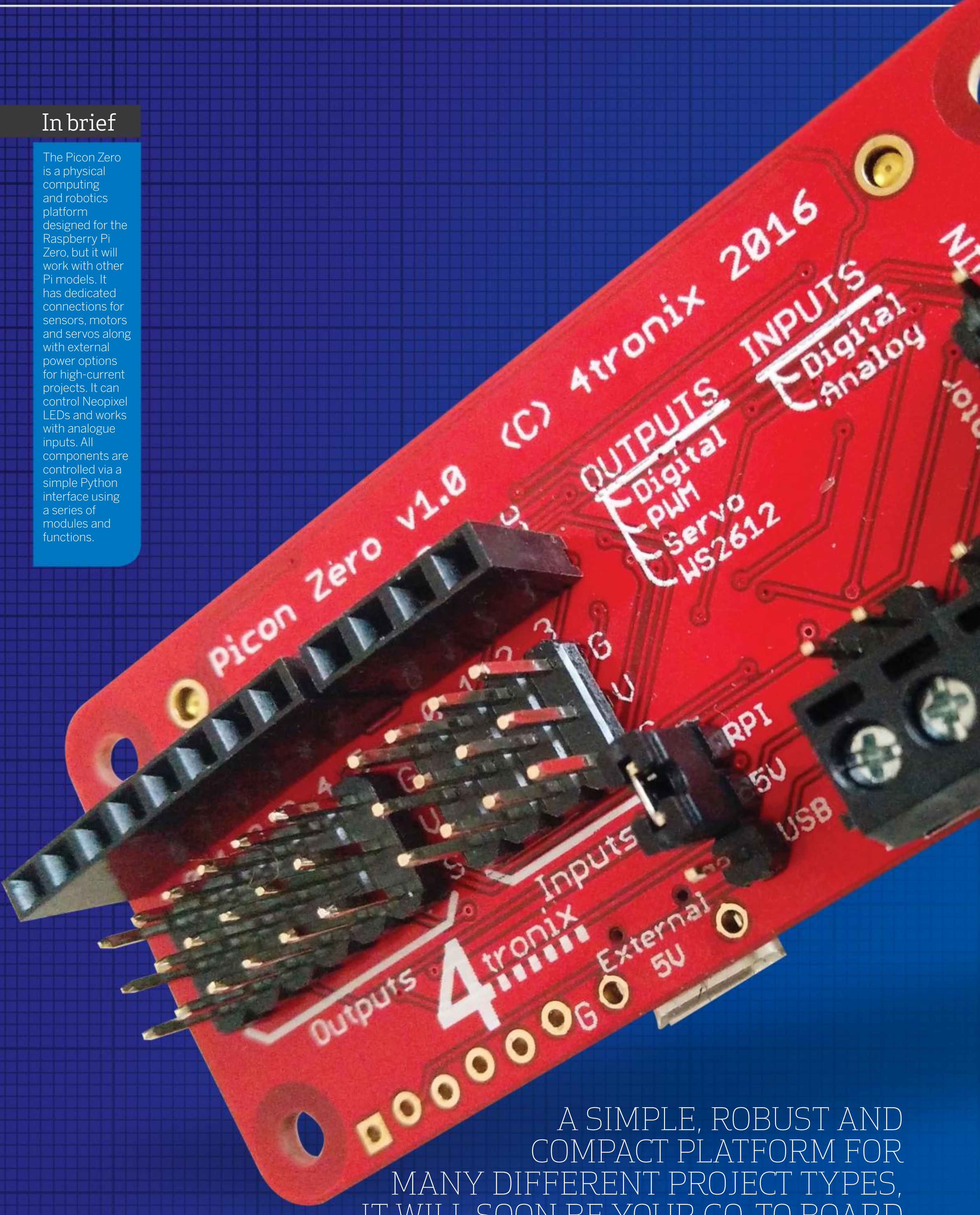
Explorer HAT Pro

Developer: Pimoroni
Web: shop.pimoroni.com/products/explorer-hat
Price: £18

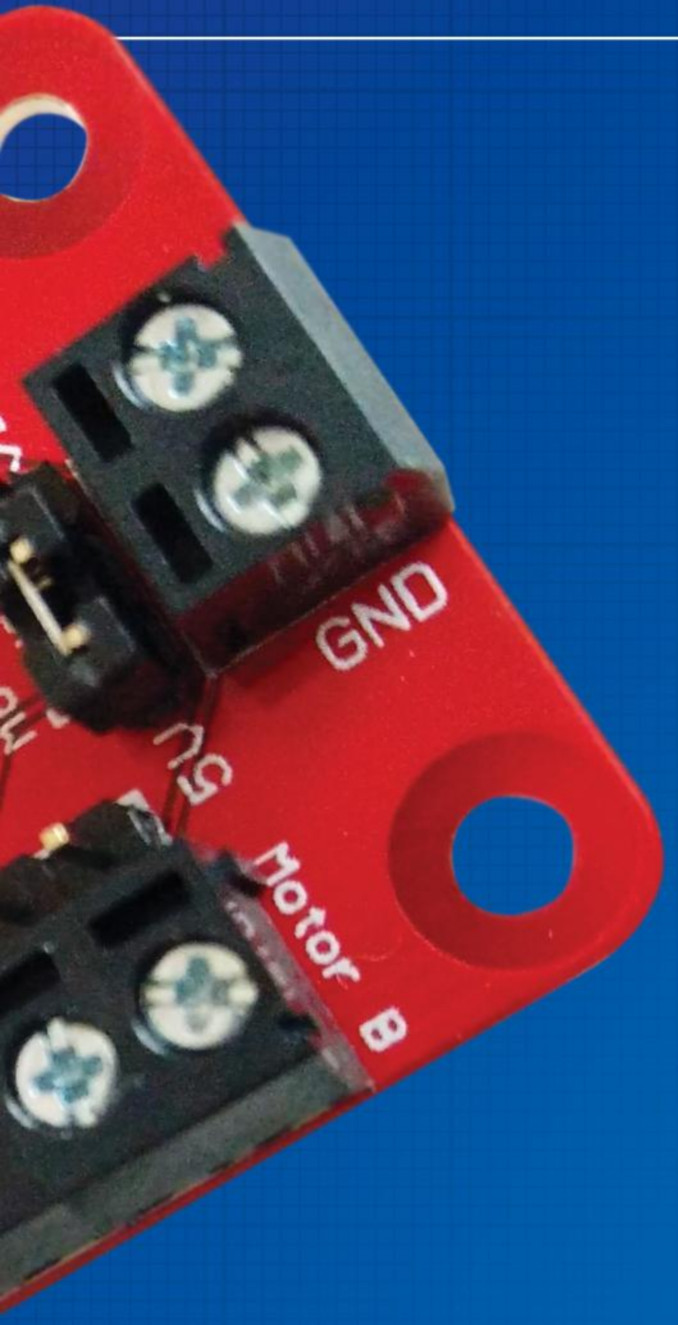
Rating 9/10

In brief

The Picon Zero is a physical computing and robotics platform designed for the Raspberry Pi Zero, but it will work with other Pi models. It has dedicated connections for sensors, motors and servos along with external power options for high-current projects. It can control Neopixel LEDs and works with analogue inputs. All components are controlled via a simple Python interface using a series of modules and functions.


Picon Zero v1.0 (C) 4tronix 2016
OUTPUTS
Digital
PWM
Servo
WS2612
INPUTS
Digital
Analog
RPI
5V
USB
Inputs
Outputs
4tronix
External
5V
G

A SIMPLE, ROBUST AND COMPACT PLATFORM FOR MANY DIFFERENT PROJECT TYPES, IT WILL SOON BE YOUR GO-TO BOARD



Picon Zero

Les Pounder might just have found the ultimate platform for his robot army

Choosing the right components for a robotics project can involve a lot of trial and error, but with the Picon Zero we might just see one board to rule them all.

The Picon Zero has been designed to match the shape of the Raspberry Pi Zero, but will also fit on top of any 40-pin GPIO Pi model. It has a set of inputs and outputs broken out as a series of male and female header pins. The bank of six outputs, labelled 0 to 5, have three pins for each device – 5V power, Ground and Signal – and can connect to servos, LEDs and even WS2812 LEDs (commonly referred to as Neopixels).

Each of these outputs can be used digitally, either on or off, or with Pulse Width Modulation (PWM), which, for example, can be used to control the brightness of an LED or control the timings of a signal. The bank of inputs follows the same physical standards as the outputs but can be used with digital inputs, such as buttons and switches, as well as analogue sensors and specialist sensors such as the DS18B20 temperature sensor.

Along the top edge of the board is a series of female connectors, the first bank of which break out a few of the spare GPIO pins from your Pi – handy if you wish to connect any other components. Just next to this bank is a special row of four pins, designed to accommodate the HC-SR04 ultrasonic sensor, commonly used to measure distance via an ultrasonic pulse much like a parking sensor. These sensors often find their way into robotics projects.

The motor controller used on the Picon Zero is a DRV8833, the same chip used for the Cam Jam Edukit 3. This dual-H-bridge controller controls two 5V motors, via two screw terminals or two male pins, at around 2A of current, enabling a powerful robot to be built. Power for your motors can come from the Raspberry Pi 5V pins, but if this proves too much for your power supply then you can swap the power supply, via a jumper, to an external USB battery pack via a handy micro USB connector on the underside of the board, or you can wire a battery box to the VIN terminals on the right of the board. The board can also control servos, small motors with a high degree of accuracy which are commonly used for precise control projects such as pan and tilt camera setups.

All of the functionality is controlled via a Python 2/3 library and it's really simple to use, thanks to a series of modules and functions that provide a human readable interface for the components and enable this board to be dropped in as a replacement for multiple boards. Think of it as a Swiss Army knife for projects.

Picon Zero provides a simple, robust and compact platform for many different project types. Other boards provide a stable platform for robotics but this really does provide everything you need to get hacking with the minimal amount of fuss. It will soon become your go-to board to get a project done. ■

Pi User Verdict

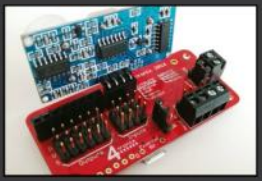
Picon Zero

Developer: 4tronix
Web: www.4tronix.co.uk
Price: £13.30

Rating 10/10

Features at a glance

External power
Powering motors can get a little too much for the Raspberry Pi, so it's handy that an external USB battery connector is included.



Ultrasonic sensor pins
The HC-SR04 sensor uses 5V logic in its signals so requires a voltage divider, which the Picon Zero has, to safely connect to the 3V logic of the GPIO pins.

CamJam EduKit 3

Inspired by the latest Star Wars movie, **Les Pounder** tries to build his own robot companion but can't figure out how to fit everything into a beachball

In brief

A robotics kit that offers an introduction to the robot creation process. The kit comes with all the parts required to build a robot but leaves the choice of chassis entirely up to the maker. Includes lessons to encourage creativity.

Robotics can become quite personal, as *Robot Wars* often proves. But the first barrier to entry is knowing what kit to purchase to get started. How much should you pay? What components will you need? For some budding makers, the issue is a lack of knowledge of skills such as soldering. Now the team behind the CamJam kits has released its own robotics kit for under £20.

The box includes everything you'd need to motorise a project. There are two large DC motors that come pre-soldered ready for use with a Pi. These fit directly into the large wheels and provide for a decently paced robot. There's also a battery box to power the motors, and a couple of sensors, notably an HC-SR04 ultrasonic sensor, which uses ultrasound to detect objects in its path, giving our robot a basic ability to "see". The other sensor is a line follower, which enables our robot to follow a bold line drawn on a surface. Other items in the box include resistors for the HC-SR04 sensor, wires, a breadboard for prototyping your robot's circuits, and a ball castor.

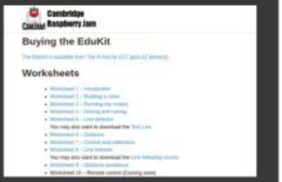
THE KIT INCLUDES EVERYTHING YOU NEED FOR A MOTORISED PROJECT – JUST ADD YOUR OWN CHASSIS FOR A UNIQUE ROBOT

What enables a robot project is a motor controller, and the CamJam kit comes with its own controller with an H bridge – a circuit that can change its polarity, enabling a motor to change direction. The DRV8833 chip that powers the controller is fully capable of controlling both of our motors. This includes full speed control thanks to PWM (Pulse Width Modulation). The motor control board can work with input power up to 10V and can power motors that require up to 2A of current. The motor control board fits over the first 26 pins of the 40-pin GPIO. This leaves a selection of pins free for use on newer Pi models but also means the board is compatible with the older 26-pin GPIO Raspberry Pi, enabling older Pis to be reused for robotics projects – an ideal reuse of them. The motor control board also provides direct access to a selection of GPIO pins to connect sensors and outputs, should the user wish to do that.

One key part of the CamJam kits is their supporting tutorials

Features at a glance

Motor Controller
The board was designed by 4tronix just for this kit and provides easy access to spare GPIO pins.

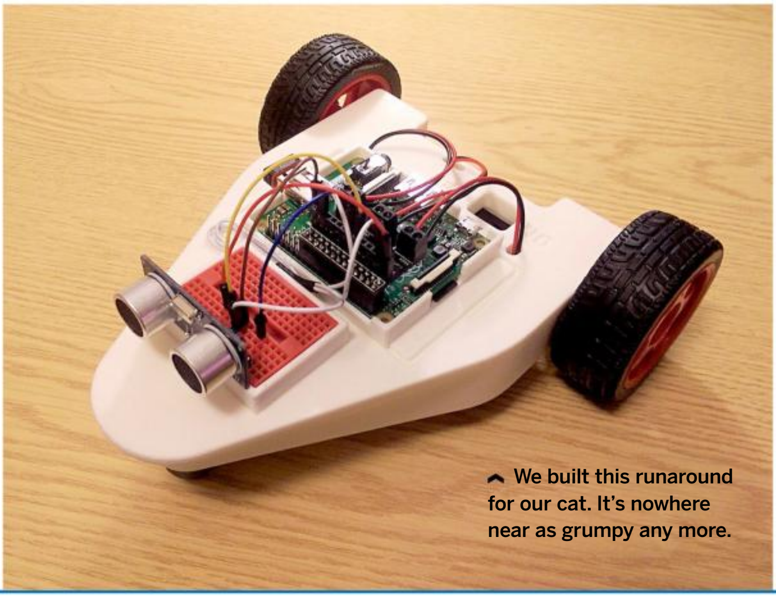


Cambridge Raspberry Jam

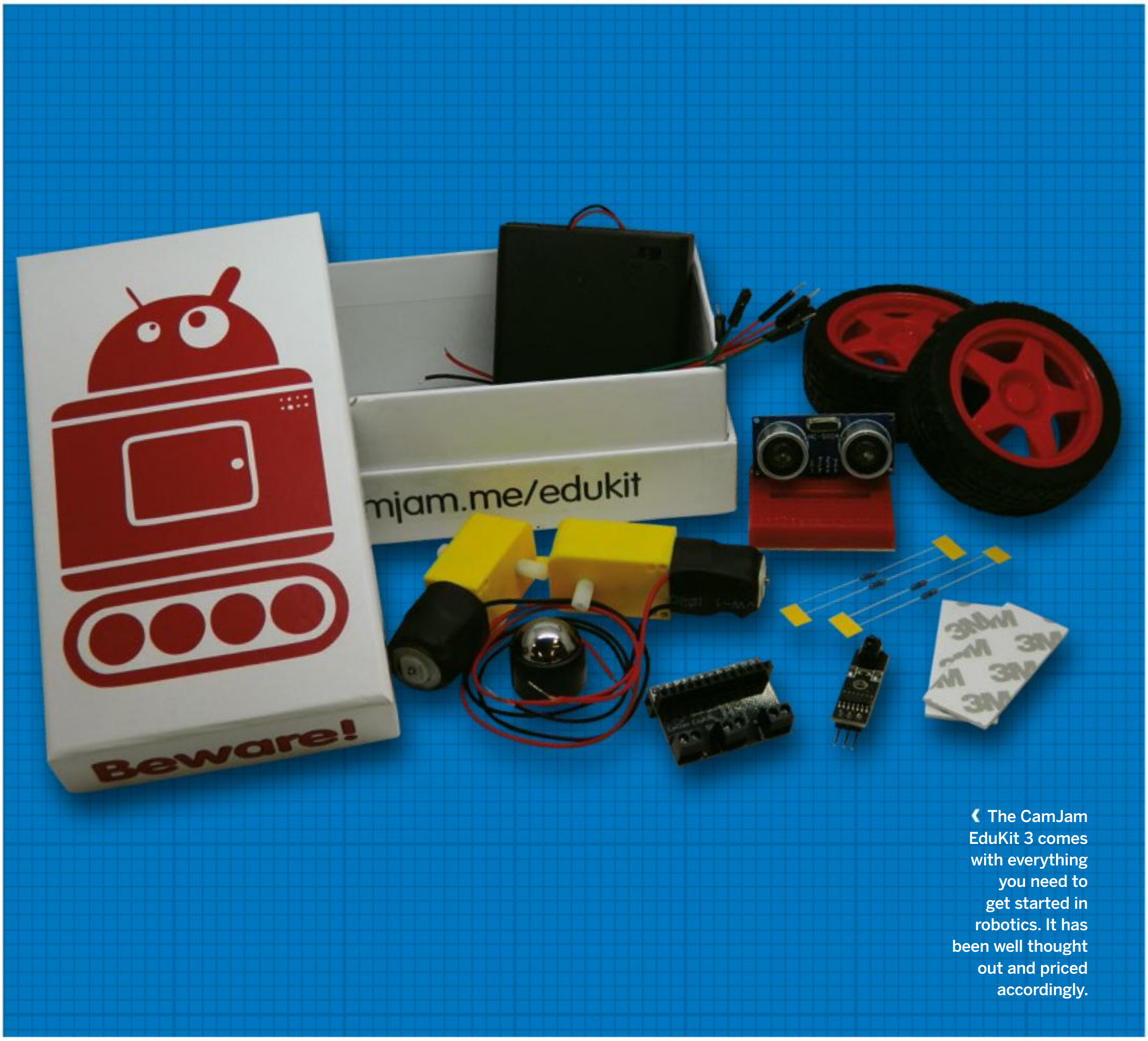
Buying the EduKit

Worksheets

Resources
The kit comes with a series of lessons that support the user through building their own robot.



We built this runaround for our cat. It's nowhere near as grumpy any more.



The CamJam EduKit 3 comes with everything you need to get started in robotics. It has been well thought out and priced accordingly.

and this kit offers the same level of quality. They come as a series of 10 projects that gradually introduce the components and process of building a robot using the kit. At present the tutorials are written using Python 3 and the RPi.GPIO library, which are easy enough to use but require a learning investment for those new to Python. Work is currently under way to port these tutorials over to the new GPIO Zero library, which offers an even easier method of working with the GPIO.

The CamJam kits are always good value, and although the CamJam Robotics kit is more expensive than its predecessors, the EduKit is still good value. It's not just about what is in the box, but rather the supporting documentation that raises this kit above others. The lessons that are built around the kit are a perfect fit for learning at a steady pace at home, and can also be used in the classroom for educators wishing to introduce robotics into the curriculum.

The beauty of this kit is in the customisation options. No chassis is provided, which helps keep the cost low but also enables you to experiment using Lego, model kits or recycled old toys to build a unique robot. In schools for example children could code the robot in Computing class and go on to build the chassis in Resistive Technologies, which has replaced shop or DT class, so this kit can provide a true cross-curricular learning experience. It's a powerful and supportive platform for experimentation and offers a gentle entry for those new to robotics. ■

Pi User Verdict

CamJam Edukit 3 Robotics

Developer: CamJam / The Pi Hut
Web: http://bit.ly/CamJam-EduKit
Price: £17

Rating 9/10

A fine addition to any desk, maker bench or coffee table (pending spouse approval).

Picade by Pimoroni

Warren Brown plugs his Pi into a retro-themed arcade gaming cabinet.

In brief...

The Picade kit from Pimoroni offers the chance to relive experience retro gaming emulation at its most authentic. It's a great option for those who prefer tinkering rather than undertaking a full-sized cab build.

The Picade offers those with fond memories of the '80s arcade scene a chance to relive those misspent days of youth. Starting it as Kickstarter project, the team at Pimoroni envisioned the Picade as an affordable all-in-one solution, one that would enable Pi users to get up and gaming with a minimum of fuss.

The kit includes a case, LCD screen, arcade PCB, speakers, buttons and control stick. It all comes in a surprisingly compact box, with the fiddly stuff compartmentalised in easy-to-reference tool boxes. Although any model Raspberry Pi can be used, we'd recommend going with the latest model, the Pi 3 model B, for best performance, and it's worth pointing out that it's not included in the box.

Build time will take just a few hours, and you'll require only a flat-head screwdriver and some elbow grease – amazingly enough, there's no need to break out the soldering kit. This being the case, the project is very family friendly, and the video instructions mean there shouldn't be any of those awkward IKEA half-built back-to-front fails. Admit it, we've all been there.

The cabinet sports a slick black powder coated finish that when coupled with the laser cut perspex bestows upon the project a level of polish few custom builds can match.

Once everything's assembled, the micro-switched controls feel responsive, with the joystick offering satisfying clicks of arcade authenticity. Sound wise the speakers bring all those iconic in-game effects to life, but quality wasn't brilliant. A substantial hum could be heard when ramping the volume up high, but at this price point you can't expect top audio quality.

The eight-inch LCD screen may sound a tad small but it didn't feel so during gaming sessions. A separate 12-inch screen is available but requires the BYO Screen Arcade Kit as opposed to the all-in-one kit featured here.

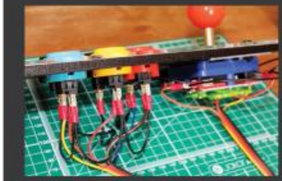
Even though the Picade is styled like a cabinet and great for playing arcade ROMs, Retropie – the go-to OS for retro emulation – can emulate a host of classic consoles such as the NES, Mega Drive or SNES. Even classic computer games for the likes of the Atari ST, Amiga and DOS can be played. Importantly, controls can be mapped via Retropie, and more specifically customised for individual titles.

But what happens when it's game over and you need your Pi back? Luckily the Picade has a handy back door and the Raspberry Pi can be easily retrieved. All-in-all the kit provides a great nostalgic fix for those who enjoy gaming's golden era.

The Picade really is an all-in-one solution at an affordable price, and an excellent way to experience those classic arcade gems of your youth. ■

RETROPIE CAN EMULATE A HOST OF CONSOLES; CONTROLS CAN BE MAPPED AND CUSTOMISED

Features at a glance



Colour coded
There's a fair bit of wiring involved, but everything is colour coded and hooked together via ready fitted crimp connectors.

Plug & play
The Pi talks to everything via a custom PCB, making connecting components a doddle. The Pi supplies all the power; making this a one cable cab. Very slick.

Pi User Verdict

Picade by Pimoroni

Developer: Pimoroni
Web: pimoroni.com
Price: £180

Rating 8/10

NEXT ISSUE

Software special!

We take a look at the best open source programs you can get for your Pi, and why you need them.

Projects

Do more with the SenseHAT, build a home robot, create a security camera for your office and more!

Coding

We continue our series of tutorials on Scratch, Python and show you even more Minecraft hacks.

Reviews

More of the latest and greatest gadgets and add-ons for your Pi. We'll turn your Pi into something special!

Look out for Pi User issue 2, on sale Friday 3 February 2017

*Contents of future issues subject to change

Pi User

Future Publishing, Quay House,
The Ambury, Bath, BA1 1UA, UK
Tel +44 (0)1225 442244
Email piuser@futurenet.com

EDITORIAL
Editor-in-Chief Graham Barlow
graham.barlow@futurenet.com
Production editor Alex Summersby
Art editors Sethnam Singh, Jamie Schildhauer, Fraser McDermott
Editorial contributors Jonni Bidwell, Les Pounder, Mayank Sharma, Robert Jones, Nate Drake, Nick Peers, Ali Jennings, Ben Everard

ADVERTISING
Commercial sales director Clare Dove
clare.dove@futurenet.com
Senior advertising manager Lara Jaggon
lara.jaggon@futurenet.com
Advertising manager Michael Pyatt
michael.pyatt@futurenet.com
Director of agency sales Matt Downs
matt.downs@futurenet.com
Ad director – Technology John Burke
john.burke@futurenet.com
Head of strategic partnerships Clare Jonik
clare.jonik@futurenet.com

MARKETING
Marketing director Sascha Kimmel
sascha.kimmel@futurenet.com

PRODUCTION AND DISTRIBUTION
Production controller Nola Cokely
Head of production UK & US Mark Constance
Distributed by Seymour Distribution Ltd, 2 East Poultry Avenue, London EC1A 9PT **Tel** 020 7429 4000
Overseas distribution by Seymour International

LICENSING
Senior Licensing & Syndication Manager Matt Ellis
matt.ellis@futurenet.com **Tel** + 44 (0)1225 442244

CIRCULATION
Trade marketing manager Juliette Winyard
Tel 07551 150 984

THE MANAGEMENT
Managing director, Magazines Joe McEvoy
Editorial director Paul Newman
Group art director Graham Dalzell

Raspberry Pi is a trademark of the Raspberry Pi Foundation. See **www.raspberrypi.org**.

LINUX is a trademark of Linus Torvalds. GNU/Linux is abbreviated to Linux throughout for brevity. All other trademarks are the property of their respective owners. Where applicable code printed in this magazine is licensed under the GNU GPL v2 or later. See **www.gnu.org/copyleft/gpl.html**.

Entire contents copyright © 2016 Future Publishing Ltd or published under licence. No part of this publication may be reproduced without written permission from the publisher. We assume all letters sent – by email, fax or post – are for publication unless otherwise stated, and reserve the right to edit contributions. All contributions to *Pi User* are submitted and accepted on the basis of non-exclusive worldwide licence to publish or license others to do so unless otherwise agreed in advance in writing. *Pi User* recognises all copyrights in this issue. Where possible, we have acknowledged the copyright holder. Contact us if we haven't credited your copyright and we will always correct any oversight. We cannot be held responsible for mistakes or misprints.

Disclaimer All tips in this magazine are used at your own risk. We accept no liability for any loss of data or damage to your computer, peripherals or software through the use of any tips or advice.

Printed in the UK by William Gibbons on behalf of Future.

Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.

Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR).
www.futureplc.com

Chief executive officer Zillah Byng-Thorne
Non-executive chairman Peter Allen
Chief financial officer Penny Ladkin-Brand
Managing director, Magazines Joe McEvoy

Tel +44 (0)1225 442 244

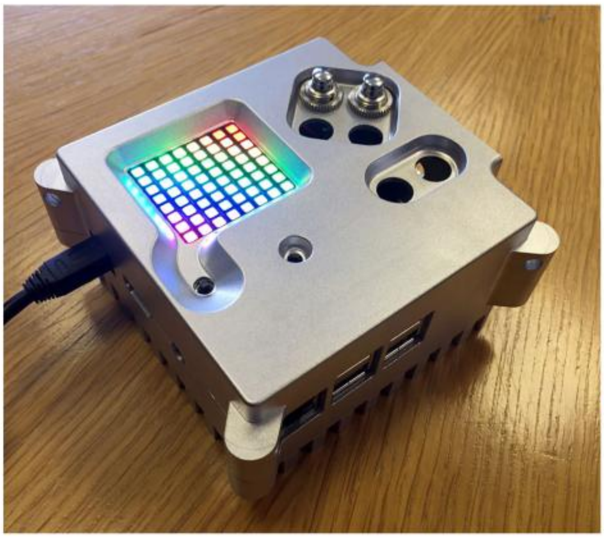
We are committed to only using magazine paper which is derived from well-managed, certified forestry and chlorine-free manufacture. Future Publishing and its paper suppliers have been independently certified in accordance with the rules of the FSC (Forest Stewardship Council).

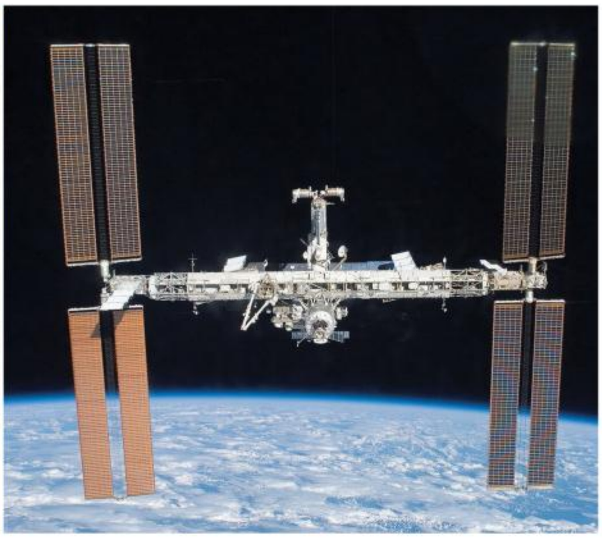
recycle
When you have finished with this magazine please recycle it.

FSC
www.fsc.org
MIX
Paper from responsible sources
FSC® C007184

The Astro Pi

Celebrating the big moments when the Raspberry Pi changed the world





It's pretty mind-blowing to know that the Raspberry Pi has been into space. It's even more amazing when you consider that there were two of them up there on the International Space Station (ISS), running programs written by UK school children.

Back in 2015 the Raspberry Pi foundation, in conjunction with the UK Space Agency, ESA and the UK Space Trade Association, ran a competition for school students to design science experiments that would run on a specially-cased Raspberry Pi called the Astro Pi, which would be taken to the ISS by British astronaut Tim Peake.

To become compatible with space travel the Astro Pi needed to be housed in a specially designed case. You see, it turns out that circuit boards with spiky edges aren't exactly ideal in a zero gravity environment where there's no kind of atmosphere on the outside. There's also the question of heat dissipation. In space the processes of convection don't happen, and with no dissipation, the warm air just stays around the CPU and bakes it, so the case needed to be really effective for thermal dissipation. (There's actually a minimal level of airflow on the ISS, which makes this possible.) There's also a rule that any surface the crew of the ISS touch must not exceed a temperature of 45 degrees. The resultant flight case was made from a single block of aluminium by a company based in Derby called Pentaxia.

> IT TURNS OUT THAT CIRCUIT BOARDS WITH SPIKY EDGES AREN'T EXACTLY IDEAL IN A ZERO GRAVITY ENVIRONMENT

The competition ran from January to July 2015 and the seven winners all had their experiments launched into space. Between February and April 2016 these experiments were run on the ISS under the supervision of Tim himself. The experiments did all sorts of cool things, using the sensors of a SenseHAT attached to the Pi. Among other things, they took pictures of the astronauts using the humidity sensor to detect them, they logged a variety of sensor measurements of the ISS interior to be used in a copy of Minecraft back on earth as a data visualisation tool, and they displayed the flag of whichever country the ISS was currently orbiting above on the LED. You can check out the results of the experiments at the Astro Pi website **https://astro-pi.org/competition/science-results/**

Most importantly, of course, the Astro Pis worked flawlessly. The experiments proved that a Raspberry Pi can run in space, and in the process a number of UK school children had the pleasure of knowing that their projects ran on the ISS. How cool is that? Maybe one day they'll be taking an Astro Pi up to the International Space Station themselves, because with the Raspberry Pi it seems that anything is possible. ■

Get the UK's best-selling Linux magazine

FREE DVD!
ALL-NEW UBUNTU 16.10
LINUX FORMAT
Get into Linux today!
65 Pages of tutorials and features
» Best Chromebooks of 2016
» Try free VPN with Opera
» Stream music with your Pi
Coding Academy: Build web apps with jQuery
SUPERCHARGE UBUNTU!
NEW RELEASE: 16.10
» Get a custom desktop
» Slick updates & upgrades
» Pro repair & system tools
OUT NOW!
Revive any PC!
Keep old devices running with Linux
LIVE DISC: BOOT IT › TRY IT › INSTALL IT!
UBUNTU 16.10
Upgrade to the latest and greatest Linux OS
» Stable, secure, speedy and free!
» Supports old and new PCs
» Get 1,000s of awesome apps
» Even faster on virtual machines
ALL NEW
32- & 64-BIT
LINUX KERNEL 4.8
CUTTING EDGE
UNITY 8 DESKTOP
NEXT-GEN PREVIEW
UPDATED APPS
GNOME 3.2 CORE
LIVE DISC: READY TO RUN
EVERYTHING YOU NEED TO GET STARTED WITH LINUX
Future

DELIVERED DIRECT TO YOUR DOOR

Order online at **www.myfavouritemagazines.co.uk**

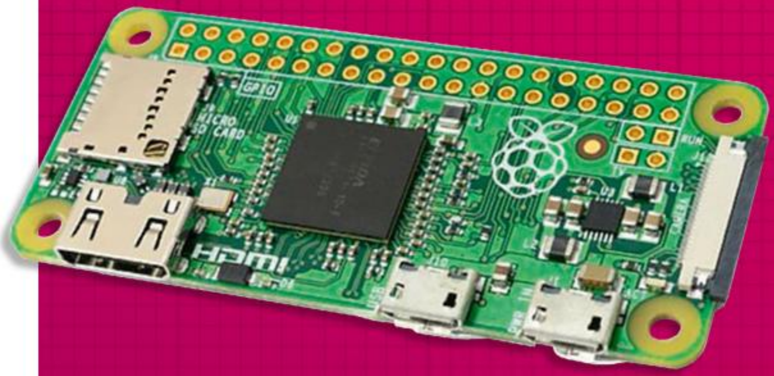
or find us in your nearest supermarket, newsagent or bookstore!

PiUser

YOUR NEW SOURCE FOR RASPBERRY PI
JOIN THE PI REVOLUTION

Discover the world of Raspberry Pi

The latest Pi projects, news and reviews

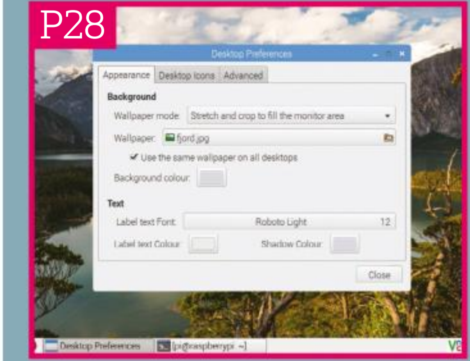


Perfect for owners of all models of the amazing Raspberry Pi and the fantastic Pi Zero

116
PAGES OF Pi GOODNESS INSIDE!

LEARN HOW TO...

P28

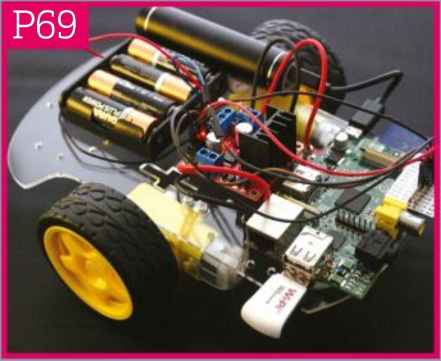


Desktop Preferences
Appearance
Desktop Icons
Advanced
Close
Desktop Preferences
[pi@raspberrypi ~]

Background
Wallpaper mode: Stretch and crop to fill the monitor area
Wallpaper: fjord.jpg
✓ Use the same wallpaper on all desktops
Background colour:
Text
Label text Font: Roboto Light 12
Label text Colour:
Shadow Colour:

Get to grips with the new Pixels Desktop

P69



Make your own Pi Zero-powered robot

P86

Program in Scratch, Python and Minecraft